04-20-00 A

Please type a plus sign (+) inside this box → [+]

PTO/SB/05 (12/97)
Approved for use through 9/30/00. OMB 0651-0032
Patent and Trademark Office: U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

jc804 U.S. PTO 04/19/00

UTILITY PATENT APPLICATION TRANSMITTAL

(Only for new nonprovisional applications under 37 CFR 1.53(b))

Attorney Docket No.	ST9-99-145	*Total Pages*	78
First Named Inventor or Application Identifier			
Shyh-Mei Ho			
Express Mail Label No.	EL487543635US		

jc525 U.S. PTO 09/552636 04/19/00

APPLICATION ELEMENTS

See MPEP chapter 600 concerning utility patent application contents.

ADDRESS TO: **Assistant Commissioner for Patents**
Box Patent Application
Washington, DC 20231

1. [X] Fee Transmittal Form
(Submit an original, and a duplicate for fee processing)

2. [X] Specification *[Total Pages* [53] *]*
(preferred arrangement set forth below)
- Descriptive title of the Invention
- Cross References to Related Application
- Statement Regarding Fed sponsored R & D
- Reference to Microfiche Appendix
- Background of the Invention
- Brief Summary of the Invention
- Brief Description of the Drawings *(if filed)*
- Detailed Description
- Claim(s)
- Abstract of the Disclosure

3. [X] Drawing(s) *(35 USC 113)* *[Total Sheets* [11] *]*

4. Oath or Declaration *[Total Pages* [4] *]*
a. [X] Newly executed (original or copy)
b. [ ] Copy from a prior application (37 CFR 1.63(d))
(for continuation /divisional with Box 17 completed)
[Note Box 5 below]
i. [ ] DELETION OF INVENTOR(S)
Signed statement attached deleting inventor(s) named in prior application, see 37 CFR 1.63(d)(2) and 1.33(b).

5. [ ] Incorporation by Reference *(useable if Box 4b is checked)*
The entire disclosure of the prior application, from which a copy of the oath or declaration is supplied under Box 4b is considered as being part of the disclosure of the accompanying application and is hereby incorporated by reference therein.

6. [ ] Microfiche Computer Program *(Appendix)*

7. Nucleotide and/or Amino Acid Sequence Submission *(if applicable, all necessary)*
a. [ ] Computer Readable Copy
b. [ ] Paper Copy (identical to computer copy)
c. [ ] Statement verifying identify of above copies

ACCOMPANYING APPLICATION PARTS

8. [X] Assignment Papers (cover sheet & document(s))

9. [ ] 37 CFR 3.73(b) Statement [ ] Power of Attorney
(when there is an assignee)

10. [ ] English Translation Document *(if applicable)*

11. [X] Information Disclosure Statement (IDS)/PTO-1449 [X] Copies of IDS Citations

12. [ ] Preliminary Amendment

13. [X] Return Receipt Postcard (MPEP 503)
(Should be specifically itemized)

14. [ ] Small Entity Statement(s) [ ] Statement filed in prior application, Status still proper and desired

15. [ ] Certified Copy of Priority Document(s)
if foreign priority is claimed)

16. [X] Other: Express Mail Certificate

17. **If a CONTINUING APPLICATION**, *check appropriate box and supply the requisite information:*

[ ] Continuation [ ] Divisional [X] Continuation-in-part (CIP) of prior application No.: 60/151,482

18. CORRESPONDENCE ADDRESS

[X] *Customer Number or Bar Code Label* 21552 *or* [ ] *Correspondence address below*

NAME	Kory D. Christensen				
ADDRESS					
CITY		STATE		ZIP	
COUNTRY		TELEPHONE		FAX	

Burden Hour Statement: This form is estimated to take 0.2 hours to complete. Time will vary depending upon the needs of the individual case. Any comments on the amount of time you are required to complete this form should be sent to the Chief Information Officer, Patent and Trademark Office, Washington, D.C. 20231. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: Assistant Commissioner for Patents, Washington, D.C. 20231.

PATENT APPLICATION
Docket No.: ST9-99-145

CERTIFICATE OF MAILING BY "EXPRESS MAIL"

"Express Mail" Mailing Label No.: EL487543635US

Date of Deposit: April 19, 2000

I hereby certify that this patent application in the name of Shyh-Mei Ho and Holger Juschkewitz for REPRESENTING IMS TRANSACTION DEFINITIONS AS XML DOCUMENTS, together with the drawings, a joint signature Declaration, Power of Attorney, and Petition, an Assignment, a Recordation Form Cover Sheet, a Fee Transmittal, a Utility Form Cover Sheet, an Information Disclosure Statement, Form PTO-1449, Cited References, and Check No. 13360 for $748.00 are being deposited with the United States Postal Service "Express Mail Post Office to Addressee" service under 37 C.F.R. § 1.10 on the date indicated above in an envelope addressed to Box Patent Application, Assistant Commissioner for Patents, Washington, D.C. 20231.

Respectfully submitted,

Kory D. Christensen
Reg. No.
Attorney for Applicant

Date: April 19, 2000

MADSON & METCALF
Gateway Tower West
15 West South Temple, Suite 900
Salt Lake City, Utah 84101
Telephone: 801/537-1700
c \corr\000418 02

Express Mailing Label No. EL487543635US

PATENT APPLICATION
Docket No. 3000.2.38
IBM No. ST9-99-145

UNITED STATES PATENT APPLICATION

of

SHYH-MEI HO

and

HOLGER JUSCHKEWITZ

for

REPRESENTING IMS TRANSACTION DEFINITIONS AS XML DOCUMENTS

MADSON & METCALF, P.C.
ATTORNEYS AT LAW
900 GATEWAY TOWER WEST
15 WEST SOUTH TEMPLE
SALT LAKE CITY, UTAH 84101

REPRESENTING IMS TRANSACTION DEFINITIONS AS XML DOCUMENTS

BACKGROUND OF THE INVENTION

Field of the Invention

The present invention relates generally to transaction processing systems. More particularly, the present invention relates to a system and method for representing IMS transaction definitions as XML documents.

Related Applications

This application claims the benefit of U.S. Application No. 60/151,482, filed August 30, 1999, for "IMS TRANSACTIONS IN XML," which is incorporated herein by reference.

Identification of Copyright

A portion of the disclosure of this patent document contains material subject to copyright protection. The copyright owner has no objection to the facsimile reproduction by anyone of the patent document or the patent disclosure, as it appears in the Patent and Trademark Office patent file or records, but otherwise reserves all copyright rights whatsoever.

MADSON & METCALF, P.C.
ATTORNEYS AT LAW
900 GATEWAY TOWER WEST
15 WEST SOUTH TEMPLE
SALT LAKE CITY, UTAH 84101

Relevant Technology

With the explosive growth of the Internet, most of the world's computer systems are now interconnected or capable of being interconnected. However, in order to share data, the systems need to understand each other's data formats. In recent years, the computer industry has evolved at such a rapid pace that systems developed only a few years apart use vastly different and incompatible formats. Such incompatibility problems tend to increase with the "age" differences of the systems.

The Information Management System (IMS) is one of the oldest and perhaps the most popular transaction processing (TP) systems. A TP system supervises the sharing of resources for concurrently processing multiple transactions, such as queries to a database. Anyone who has ever used an ATM, rented a car, or booked a flight, has probably used IMS.

IMS was developed by IBM the 1960's as a inventory tracking system for the U.S. moon landing effort. Today, interfacing IMS with newer systems, particularly with systems of different manufactures over the Internet, is problematic.

As illustrated in Figure 1, an IMS 10 typically includes two major components: an IMS Transaction Monitor (IMS/TM) 12, which is responsible for scheduling, authorization, presentation services and operational functions, and a hierarchical database 14, DL/I. Both components are independently configurable. For example the IMS/TM 12 can use a relational database, such as DB/2, rather than

MADSON & METCALF, P.C.
ATTORNEYS AT LAW
900 GATEWAY TOWER WEST
15 WEST SOUTH TEMPLE
SALT LAKE CITY, UTAH 84101

DL/I. The various components of an IMS 10 communicate via the MVS operating system 16.

As illustrated Figure 2, the architecture of IMS is divided into four regions. For example, a Message Processing Region (MPR) 20 is used to execute message-driven applications 18. Execution of applications 18 in this region 20 is triggered by incoming messages, such as those received from a terminal.

By contrast, applications 18 in a Batch Message Processing (BMP) 22 region are not message driven. They are usually scheduled to run at times of low system activity, such as nights and weekends. Typically, such applications 18 perform a number of predefined operations, after which they immediately terminate.

An Interactive Fast Path (IFP) 24 region allows fast and simple requests to the database 14. Applications 18 operating in the IFP 24 bypass the normal scheduling mechanism, providing relatively fast response times. In general, IFP applications 18 stay resident even if they are not needed.

An IMS Control Region (IMSCTL) 26 is responsible for all TP tasks, as well as for controlling all dependent regions (MPR 20, BMP 22, and IFP 24). Essentially, the IMS Control Region 26 has three main responsibilities: telecommunications, message scheduling, and logging/restart.

For example, IMSCTL 26 controls the connected terminals 28 (illustrated in Figure 3), receiving/sending messages from/to the terminals 28. Moreover,

MADSON & METCALF, P.C.
ATTORNEYS AT LAW
900 GATEWAY TOWER WEST
15 WEST SOUTH TEMPLE
SALT LAKE CITY, UTAH 84101

IMSCTL 26 logs all transactions in order to provide the capability of undoing non-committed transactions in the event of a system failure.

In addition, every time IMSCTL 26 receives a message from a terminal 28, it schedules an application 18 to process the message. IMSCTL 26 identifies the desired application 18 and puts the message in the application's message queue 30. The application 18 processes the requests in its message queue 30 and responds to the originating terminal 28 by placing the response in the terminal's message queue 32.

As illustrated in Figure 4, an IMS 10 obtains all of its information about the structure and behavior of its components (applications, databases, transactions, etc.) from macro statements 34 (hereinafter "macros"). Certain macros 34 are referred to as "transaction definitions" 35 because they define how transactions are processed.

For example, as shown in Figure 4, an application (APPLCTN) macro 36 defines the behavior of a particular IMS application 18. An APPLCTN macro 36 exists for each application 18 in the IMS 10, and defines, for example, the application's name, resource requirements, and appearance.

An APPLCTN macro 36 is followed by a zero or more (TRANSACT) macros 38, which define the various transactions applicable to the application 18. A TRANSACT macro 38 specifies the appearance of a transaction to be performed by an application 18, identifying whether the transaction is IMS exclusive, IMS Fast

MADSON & METCALF, P.C.
ATTORNEYS AT LAW
900 GATEWAY TOWER WEST
15 WEST SOUTH TEMPLE
SALT LAKE CITY, UTAH 84101

Path potential or IMS Fast Path exclusive. Furthermore, a TRANSACT macro 38 specifies the transaction code that causes the application 18 named in the APPLCTN macro 36 to be scheduled for execution in an IMS processing region.

Currently, IMS is only capable of processing transactions previously defined by the APPLCNT and TRANSACT macros 36, 38. A user may not initiate arbitrary transactions, such queries of the database 14, that have not been previously defined. Moreover, in order to change the above-described macros, one must initiate a process called "system generation," which necessitates shutting down the IMS 10 for a period of time.

In addition, the above-described IMS macros 36, 38 have a proprietary format, which is a detriment in interfacing with remotely located systems from different vendors. Currently, the dominant Internet format is the HyperText Markup Language (HTML), a variant of the eXtensible Markup Language (XML). Providing a technique for delivering IMS transactions definitions to an IMS 10 using interchangeable documents, such as XML documents, would be a first step in being able to initiate arbitrary IMS transactions over the Internet.

Accordingly, what is needed is a system and method for representing IMS transaction definitions in an interchangeable format, such as XML. What is also needed is a system and method for communicating with an IMS 10 using XML documents. In addition, what is needed is a system and method for creating a

MADSON & METCALF, P.C.
ATTORNEYS AT LAW
900 GATEWAY TOWER WEST
15 WEST SOUTH TEMPLE
SALT LAKE CITY, UTAH 84101

Document Type Definition (DTD) for use by a parser in converting between IMS transaction definitions and XML documents.

MADSON & METCALF, P.C.
ATTORNEYS AT LAW
900 GATEWAY TOWER WEST
15 WEST SOUTH TEMPLE
SALT LAKE CITY, UTAH 84101

SUMMARY OF THE INVENTION

The present invention solves many or all of the foregoing problems by providing a system and method for communicating with an Information Management System (IMS) using eXtensible Markup Language (XML) documents.

In one aspect of the invention, a method includes the steps of receiving a document comprising an IMS transaction definition encoded in XML; obtaining a Document Type Definition (DTD) specifying rules for decoding the IMS transaction definition; parsing the XML document using the DTD to decode the IMS transaction definition; and providing the decoded IMS transaction definition to the IMS.

In another aspect of the invention, a method includes the steps of modeling an IMS transaction definition in a Universal Modeling Language (UML) to produce a UML object model; and processing the UML object model using an XML Metadata Interchange (XMI) utility to create the DTD.

In yet another aspect of the invention, a method includes the steps of obtaining an IMS transaction definition; obtaining a Document Type Definition (DTD) specifying rules for encoding the IMS transaction definition; and parsing the IMS transaction definition with the DTD to encode the IMS transaction definition in an XML document.

In still another aspect, a system includes a document reception module configured to receive a document comprising an IMS transaction definition encoded in XML; a parser configured to obtain a Document Type Definition (DTD) specifying

MADSON & METCALF, P.C.
ATTORNEYS AT LAW
900 GATEWAY TOWER WEST
15 WEST SOUTH TEMPLE
SALT LAKE CITY, UTAH 84101

rules for decoding the IMS transaction definition and further configured to parse the XML document using the DTD to decode the IMS transaction definition; and an IMS interface module configured to provide the decoded IMS transaction definition to the IMS.

In another aspect of the invention, a system includes a modeling tool configured to model an IMS transaction definition in a Universal Modeling Language (UML) to produce a UML object model; and an XML Metadata Interchange (XMI) utility configured to process the UML object model to generate the DTD and a number of access classes.

These and other objects, features, and advantages of the present invention will become more fully apparent from the following description and appended claims, or may be learned by the practice of the invention as set forth hereinafter.

MADSON & METCALF, P.C.
ATTORNEYS AT LAW
900 GATEWAY TOWER WEST
15 WEST SOUTH TEMPLE
SALT LAKE CITY, UTAH 84101

BRIEF DESCRIPTION OF THE DRAWINGS

The present invention is more fully disclosed in the following specification, with reference to the accompanying drawings, in which:

Figure 1 is a schematic block diagram of an Information Management System (IMS);

Figure 2 is a schematic block diagram of IMS processing regions;

Figure 3 is a schematic block diagram of message processing within an IMS;

Figure 4 is a schematic block diagram of IMS macros including two IMS transaction definitions;

Figure 5 is a schematic block diagram of a system for communicating with an IMS using eXtensible Markup Language (XML) documents according to an embodiment of the invention;

Figure 6 is a schematic block diagram of a system for generating XML documents from IMS transaction definitions according to an embodiment of the invention;;

Figure 7 is a schematic block diagram of a system for generating a parser, a number of access classes, and a Document Type Definition (DTD) for IMS transaction definitions according to an embodiment of the invention;

Figure 8 is a schematic block diagram of a Universal Modeling Language (UML) object model representing IMS transaction definitions according to an embodiment of the invention;

MADSON & METCALF, P.C.
ATTORNEYS AT LAW
900 GATEWAY TOWER WEST
15 WEST SOUTH TEMPLE
SALT LAKE CITY, UTAH 84101

Figure 9 is a schematic flowchart of a method for communicating with an IMS using XML documents according to an embodiment of the invention;

Figure 10 is a schematic block diagram showing a conversion between an XML document and an IMS transaction definition according to an embodiment of the invention;

Figure 11 is a schematic flowchart of a method for converting an IMS transaction definition into an XML document according to an embodiment of the invention;

Figure 12 is a schematic block diagram showing a conversion between an IMS transaction definition into an XML document according to an embodiment of the invention; and

Figure 13 is a schematic flowchart of a method for generating a parser, a number of access classes, and a DTD for IMS transaction definitions according to an embodiment of the invention.

MADSON & METCALF, P.C.
ATTORNEYS AT LAW
900 GATEWAY TOWER WEST
15 WEST SOUTH TEMPLE
SALT LAKE CITY, UTAH 84101

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

Certain presently preferred embodiments of the invention are now described with reference to the Figures, where like reference numbers indicate identical or functionally similar elements. The components of the present invention, as generally described and illustrated in the Figures, may be implemented in a variety of configurations. Thus, the following more detailed description of the embodiments of the system and method of the present invention, as represented in the Figures, is not intended to limit the scope of the invention, as claimed, but is merely representative of presently preferred embodiments of the invention.

Throughout the following description, various system components are referred to as "modules," "units," "utilities," "tools," or the like. In certain embodiments, these components may be implemented as software, hardware, firmware, or any combination thereof.

For example, as used herein, a module may include any type of computer instruction or computer executable code located within a memory device and/or transmitted as electronic signals over a system bus or network. An identified module may include, for instance, one or more physical or logical blocks of computer instructions, which may be embodied within one or more objects, procedures, functions, or the like.

The identified modules need not be located physically together, but may include disparate instructions stored at different memory locations, which together

MADSON & METCALF, P.C.
ATTORNEYS AT LAW
900 GATEWAY TOWER WEST
15 WEST SOUTH TEMPLE
SALT LAKE CITY, UTAH 84101

implement the described logical functionality of the module. Indeed, a module may include a single instruction, or many instructions, and may even be distributed among several discrete code segments, within different programs, and across several memory devices.

Figure 5 illustrates a schematic block diagram of a system 40 for communicating with an Information Management System (IMS) 10 using interchangeable documents according a presently preferred embodiment of the invention. The system 40 may include a Web server 46, which may receive a specially-encoded document 44 from a Web browser 42 via the Internet 48.

Various Web servers 46 may be used, such as the Windows NT Server™, available from Microsoft Corporation. Likewise, the Web browser 42 may be implemented using a conventional navigation tool for accessing the World Wide Web (WWW), such as Microsoft Internet Explorer™ or Netscape Navigator™.

The document 44 may be encoded to represent one or more IMS transaction definitions 35, such as APPLCTN or TRANSACT macros 36, 38, using an open document format, such as the eXtensible Markup Language (XML). Techniques for encoding IMS transaction definitions 35 in XML are described more fully hereafter. XML is similar to the HyperText Markup Language (HTML), but provides the ability to define custom tags by means of Document Type Definitions (DTDs).

In various embodiments, the system 40 also includes an IMS gateway 50, which may provide an interface between the Web server 46 and the IMS 10. The

MADSON & METCALF, P.C.
ATTORNEYS AT LAW
900 GATEWAY TOWER WEST
15 WEST SOUTH TEMPLE
SALT LAKE CITY, UTAH 84101

IMS gateway 50 may include an XML to Macro parser 52, which may parse the XML document 44 using a specialized DTD 54 to decode the IMS transaction definition 35. A system and method for generating the DTD 54 and the parser 52 is described more fully hereafter. After the transaction definition 35 is decoded, the IMS gateway 50 may provide the same to the IMS 10.

In alternative embodiments, the system 40 may be used in reverse to generate an XML document 44 from an IMS transaction definition 35, as illustrated Figure 6. For example, a transaction definition 35 may be obtained from the IMS 10 or another source. Thereafter, a Macro to XML parser 53 within the IMS gateway 50 may use the DTD 54 to encode the transaction definition 35 as an XML document 44. In certain embodiments, two different parsers 52, 53 are used, one for encoding and one for decoding a transaction definition 35, although the invention is not limited in this respect. After the XML document 44 is generated, the Web server 46 may transmit the same to a Web browser 42 or the like via the Internet 48.

Figure 7 illustrates a system 60 for generating the DTD 54 and a number of access classes 76 and for creating the parsers 52, 53 based on the DTD 54 and the access classes 76. In various embodiments, the system 60 includes a Uniform Modeling Language (UML) modeling tool 62. The UML modeling tool 62 may be used to produce a UML object model 64 to represent the transaction definitions 35. In one embodiment, the UML modeling tool 62 is Rational Rose™, a visual modeling tool available from Rational Software. Appendix A includes a UML Model Report

MADSON & METCALF, P.C.
ATTORNEYS AT LAW
900 GATEWAY TOWER WEST
15 WEST SOUTH TEMPLE
SALT LAKE CITY, UTAH 84101

generated by Rational Rose, as well as a description of various UML classes according to an embodiment of the invention.

UML is a language for specifying, visualizing, constructing, and documenting software systems, as well as business models and the like. UML is capable of representing the static, dynamic, environmental, and organizational aspects of almost any system.

Figure 8 illustrates a UML object model 64 of the APPLCTN and TRANSACT macros 36, 38, as displayed by the UML modeling tool 62. In one embodiment, the model 64 includes an *ApplicationControlMacro* object 66 to represent the APPLCTN macro 36. Each parameter of the APPLCTN macro 36 may be represented by an attribute 67 of the *ApplicationControlMacro* object 66. For example, the PSBName attribute represents a parameter in the APPLCTN macro 36 for the Program Specification Block (PSB) name.

As illustrated, the *ApplicationControlMacro* object 66 may be linked to zero or more *TransactionControlMacro* objects 68. In one embodiment, a *TransactionControlMacro* object 68 represents the TRANSACT macro 38, each attribute 69 of the object 68 corresponding to a parameter of the macro 38.

In addition, the UML object model 64 may include a number of enumeration stereotypes 70, which are type definitions of attributes 67, 69 having a fixed set of values. For example, an attribute 67 of the type, TscheduleType, may only have a value of "Serial" or "Parallel" in certain embodiments. Likewise, the UML object

MADSON & METCALF, P.C.
ATTORNEYS AT LAW
900 GATEWAY TOWER WEST
15 WEST SOUTH TEMPLE
SALT LAKE CITY, UTAH 84101

model 64 may include a number of primitive stereotypes 72, which represent the basic variable types of the model 64, such as strings, unsigned integers, and the like.

Referring again to Figure 7, the system 60 may also include an XML Metadata Interchange (XMI) utility 74, such as XMI Toolkit™, available from IBM. XMI is an open standard released by the Object Management Group (OMG) for simplifying the exchange of data and metadata between different products from different vendors. IBM's XMI Toolkit is written entirely in Java and offers interfaces to facilitate incorporation into other projects or products.

In various embodiments, the XMI utility 74 automatically generates the DTD 54 from the UML object model 64. An example of a DTD 54 generated by the XMI utility 74 according to an embodiment of the invention is provided in Appendix B. In addition, the XMI utility 74 may create a plurality of Java XMI document access classes 76, which are used by the parsers 52, 53 to parse and transform the document 44 based on the generated DTD 54. The access classes 76 provide an application programming interface (API) to allow the parsers 52, 53 to export a transaction definition 35 to an XML document 44 and to import an XML document 44 to a transaction definition 35.

The system 60 may also include a parser generator 78, such as the Java Compiler Compiler (JavaCC) 1.1, available from Sun Microsystems, although a variety of parser generators 78 may be used. JavaCC converts a grammar specification into a Java program that can recognize matches to the grammar. In

MADSON & METCALF, P.C.
ATTORNEYS AT LAW
900 GATEWAY TOWER WEST
15 WEST SOUTH TEMPLE
SALT LAKE CITY, UTAH 84101

certain embodiments, the parser generator 78 is only used in the creation of the parser 53 (Macro to XML), since the XMI access classes 76 generated by the XMI utility 74 may be used to directly import the entire content of the document 44 by invoking a single method. However, other techniques for parser generation may be used in alternative embodiments.

Referring now to Figure 9, a schematic flowchart illustrates a method 80 for communicating with an IMS 10 using XML documents 44. The method may begin by receiving 82 a document 44 at an IMS gateway 50. The document 44 may be received, for example, from a Web server 46 attached to the Internet 48.

After the document 44 is received 82, the method 80 may continue by obtaining 84 a DTD 54 for IMS transaction definitions 35, such as the DTD 54 illustrated in Appendix B. Thereafter, the method may continue by parsing 86 the document 44 using the DTD 54 to decode the IMS transaction definition 35, and providing 88 the same to the IMS 10.

Figure 10 illustrates the above-described process. For example, a portion of an XML-encoded document 44 representing an APPLCTN macro 36 is shown. A parser 52 obtains the DTD 54 and parses the document 44 to decode the macro 36, i.e. "APPLCTN PSB=ABCD,PGMTYPE=(TP,1)."

Figure 11 illustrates a method 90 for generating an XML document 44 from an IMS transaction definition 35 according to an embodiment of the invention. The

MADSON & METCALF, P.C.
ATTORNEYS AT LAW
900 GATEWAY TOWER WEST
15 WEST SOUTH TEMPLE
SALT LAKE CITY, UTAH 84101

method 90 may begin by obtaining 92 an IMS transaction definition 35 from an IMS 10 or another source.

After the transaction definition 35 is obtained 92, the method 90 may continue by obtaining 94 a DTD 54 for IMS transaction definitions 35, as illustrated in Appendix B. Thereafter, the method 90 may continue by parsing 96 and transforming the IMS transaction definition 35 using the DTD 54 to encode the same as an XML document 44. Finally, the XML document 44 may be sent 98 to an XMI-enabled device, such as a Web browser 42 or the like.

Figure 12 illustrates the above-described process. A subset of an APPLCTN macro 36 is shown, i.e. "APPLCTN PSB=ABCD,FPATH=30720." The parser 53 obtains the DTD 54 and parses and transforms the macro 36 to encode the same as an XML document 44, as illustrated.

Figure 13 illustrates a method 100 for generating a DTD 54 and a parser 53 in accordance with an embodiment of the invention. The method 100 may begin by modeling 102 an IMS transaction definition 35 in a Universal Modeling Language (UML) to produce a UML object model 64. As previously noted, a visual modeling tool, such as Rational Rose, may be used for this purpose.

After the transaction definition 35 is modeled 102, the method 100 may continue by processing 104 the UML object model 64 using an XMI utility 74, such as IBM's XMI Toolkit, to generate the DTD 54 and various document access classes 76.

MADSON & METCALF, P.C.
ATTORNEYS AT LAW
900 GATEWAY TOWER WEST
15 WEST SOUTH TEMPLE
SALT LAKE CITY, UTAH 84101

In certain presently preferred embodiments, the method 100 may continue by processing the DTD 54 using a parser generator 78, such as the Java Compiler Compiler (JavaCC), in order to generate the parser 53.

Based on the foregoing, the present invention offers a number of advantages over conventional IMS interface systems. For example, the present invention communicates with an IMS 10 using openly interchangeable documents, such as XML documents 44. The documents 44 may comply with the latest XMI standard, which enables an IMS 10 to exchange data with a variety of XMI-enabled devices, such as Web browsers 42 and the like. Accordingly, the present invention overcomes a first hurdle in the goal of facilitating arbitrary IMS transactions via the Internet.

Importantly, XML documents 44 may be easily converted for display on a variety of computing platform using the emerging XML Style Language (XSL) standard. As such, the XMI to IMS interface is capable of replacing all other interfaces between IMS and products from other vendors.

The present invention may be embodied in other specific forms without departing from its scope or essential characteristics. The described embodiments are to be considered in all respects only as illustrative and not restrictive. The scope of the invention is, therefore, indicated by the appended claims rather than by the foregoing description. All changes which come within the meaning and range of equivalency of the claims are to be embraced within their scope.

MADSON & METCALF, P.C.
ATTORNEYS AT LAW
900 GATEWAY TOWER WEST
15 WEST SOUTH TEMPLE
SALT LAKE CITY, UTAH 84101

What is claimed is:

MADSON & METCALF, P.C.
ATTORNEYS AT LAW
900 GATEWAY TOWER WEST
15 WEST SOUTH TEMPLE
SALT LAKE CITY, UTAH 84101

MADSON & METCALF, P.C.
ATTORNEYS AT LAW
900 GATEWAY TOWER WEST
15 WEST SOUTH TEMPLE
SALT LAKE CITY, UTAH 84101

APPENDIX A

UML Model Report and Class Descriptions

Overview

The following is a UML model report generated by Rational Rose, as well as descriptions of various UML classes according to one embodiment of the invention.

Content

SubComponents

None

Classes

ApplicationControlMacro

TransactionControlMacro

TResidency

TLanguage

TScheduleType

TProgramType

Boolean

TMode

MessageType

String

TSysID

TFPBuffer

TClass

TPriority

TScheduleOption

TParallelLimit

TMaxRegions

ProcessLimitType

UnsignedInteger

TSpaSize

TSegment

TResponse

Classes

ApplicationControlMacro Class

Introduction

TheApplicationControlMacro defines the program resource requirements for application programs that run under the control of the IMS DB/DC environment, as well as for applications that access databases through DBCTL. When combined with one or more TransactionControlMacros, the ApplicationControlMacro defines the scheduling and resource requirements for an application program. On the ApplicationControlMacro, one only describes programs that operate in message processing regions, Fast Path message-driven program regions, batch message processing regions, or CCTL threads. The ApplicationControlMacro is also used to describe application programs that operate in batch processing regions.

Base Classes

None

Attributes

ResidencyOption: TResidency

The positional parameter Resident specifies that the PSB associated with the application program is to be made resident during system initialization. Resident and DynamicOption (dynamic PSB option) are mutually exclusive. DynamicOption and ScheduleType=Parallel are also mutually exclusive.

Neither Resident nor DynamicOption is the default parameter. Rather, if neither Resident nor DynamicOption is selected on the ApplicationControlMacro statement, IMS system initialization causes a BLDL to be performed on the PSB associated with the application being defined. The PSB is not made resident (that is, loaded from the ACBLIB) until the application is scheduled.

isFPath: Boolean = False

Specifies whether the application is a Fast Path exclusive application program. If True, the ProgramType parameter is invalid and TransactionControlMacro.isWaitForInput=True is forced.

FPathBufferSize: TFPBuffer

Determines the EMH buffer size required to run the transaction and overrides the EML execution parameter.

MADSON & METCALF, P.C.
ATTORNEYS AT LAW
900 GATEWAY TOWER WEST
15 WEST SOUTH TEMPLE
SALT LAKE CITY, UTAH 84101

GeneratedPSBName: String

Defining this attribute causes the scheduling process of all environments to generate a PSB containing an I/O PCB and an alternate modifiable PCB. With this attribute coded, one does not need to perform the PSBGEN and the ACBGEN, thus eliminating I/O to the ACBLIB.

GeneratedPSBName generates an I/O PCB named IOPCBbbb; The modifiable, alternate PCB is named TPPCB1bb. With an alternate modifiable PCB, an application can use the CHNG call to change the output destination and send output to a destination other than the input destination.

The GeneratedPSBName and the PSBName attributes are mutually exclusive.

GeneratedPSBLanguage: TLanguage = Assembler

Defines the language interface of the application program.

This attribute may only be valid if the GeneratedPSBName is specified.

ProgramType: TProgramType = TeleProcessing

Describes the type of application program being defined. The default, TeleProcessing, specifies that IMS schedules the program when messages processed by the program exist in the system. A program defined as Batch can use DL/I in the IMS control program system region and can refer to the message queues. If Batch is coded, all TransactionControlMacros statements that follow are assigned normal and limit priority values of zero.

ProgramClass: TClass = 1

This attribute specifies the class to which the transaction codes specified in the TransactionControlMacro statements are to be assigned.

The value should not exceed the value given (by specification or default) on the MAXCLAS keyword of the IMSCTRL macro.

If the transaction code class is defined in the individual TransactionControlMacro statement, this parameter is ignored.

The numeric class parameter should not be specified if ApplicationControlMacro.isFPath=True is specified.

PSBName: String

Specifies the name of the PSB associated with this application program definition. Each local PSB name must be unique. A remote and a local application can each be defined as having the same PSB name. This may be required in order to dynamically reassign a transaction from remote to local processing.

The GeneratedPSBName and the PSBName attributes are mutually exclusive. The PSBName is required, if no GeneratedPSBName is specified.

ScheduleType: TScheduleType = Serial

MADSON & METCALF, P.C.
ATTORNEYS AT LAW
900 GATEWAY TOWER WEST
15 WEST SOUTH TEMPLE
SALT LAKE CITY, UTAH 84101

Specifies the application program can be scheduled into more than one message region or batch message region simultaneously..

ScheduleType and ResidencyOption=DynamicOption are mutually exclusive.

SysIDRemote: TSysID

Specifies in the multiple IMS system configuration, the system identification (SysID) of the remote system (that system on which the application executes).

If the ApplicationControlMacro.SysID parameters are defined, all other parameters of the ApplicationControlMacro except the PSB are ignored. If the TransactionControlMacro.SysID parameters are defined, the ApplicationControlMacro.SysID parameters are ignored. This parameter is invalid, if ApplicationControlMacro.isFPath=True is coded.

If this attribute is specified the SysIDLocal attribute must be specified, too.

SysIDLocal: TSysID

Specifies in the multiple IMS system configuration, the system identification (SysID) of the local system (the originating system to which the response are returned).

If the ApplicationControlMacro.SysID parameters are defined, all other parameters of the ApplicationControlMacro except the ApplicationControlMacro.PSB are ignored. If the TransactionControlMacro.SysID parameters are defined, the ApplicationControlMacro.SysID parameters asr ignored. This attribute is invalid, if ApplicationControlMacro.isFPath=True is coded.

If this attribute is specified the SysID_Remote attribute must be specified, too.

Inherited Attributes

Associations

TRANSACT with TransactionControlMacro

TransactionControlMacro Class

Introduction

The TRANSACT macro statement is used one or more times with each ApplicationControlMacro statement to identify a transaction as IMS exclusive, IMS Fast Path potential or IMS Fast Path exclusive. It specifies the transaction codes that cause the application program named in the preceding ApplicationControlMacro to be scheduled for execution in an IMS message processing region. It also provides the IMS control program with information that influences the application program scheduling algorithm. It can define a message editing routine.

Base Classes

MADSON & METCALF, P.C.
ATTORNEYS AT LAW
900 GATEWAY TOWER WEST
15 WEST SOUTH TEMPLE
SALT LAKE CITY, UTAH 84101

None

Attributes

TransactionCode: String

Specifies the transaction code associated with this TransactionControlMacro statement.

isDCLogWriteAhead: Boolean

Specifies whether IMS should perform log write-ahead for recoverable, nonresponse mode input messages and transactions output messages. If not specified in the TransactionControlMacro, the default is the DCLogWriteAhead parameter in the IMSCTRL macro. The DCLogWriteAhead parameter in the TransactionControlMacro overrides the IMSCTRL macro parameter for this transaction.

toUpperCase: Boolean = True

Specifies whether (True) or not (False) the input data is to be translated to uppercase before beeing presented to the processing program.

Specifying True for VTAM terminals prevents the transmission of embedded device control characters.

Specifying the attribute requires specifying the EditRoutine attribute.

EditRoutine: String

This attribute allows a user to a transaction input edit routine that edits messages prior to the program receiving the message. The specified edit routine (load module) must reside on the USERLIB data set prior to IMS system definition stage 2 execution. This routine cannot be the same as the one that is used on a LINEGRP or TYPE EDIT parameter.

This attribute is invalid on a Fast Path exclusive transaction.

isFPath: Boolean = False

Specifies whether the transaction code is a potential candidate for Fast Path processing. Specifying True is effective only when specified on a TransactionControlMacro that follows an ApplicationControlMacro statement that does not specify isFPath=True; otherwise, the operand is ignored.

Fast Path potential transactions must be processed by a user edit/routing exit to determine whether the transaction is actually to be processed by IMS Fast Path. If it is to be processed by IMS Fast Path, the edit/routing exit associates the transaction with a routing code. This routing code identifies which Fast Path application program is to process the transaction.

FPathBufferSize: TFPBuffer

Determines the EMH buffer size required to run the transaction and overrides the EML execution parameter.

MADSON & METCALF, P.C.
ATTORNEYS AT LAW
900 GATEWAY TOWER WEST
15 WEST SOUTH TEMPLE
SALT LAKE CITY, UTAH 84101

isInquiry: Boolean = False

Specifies whether the transaction is an inquiry transaction.

isRecover: Boolean = True

Specifies whether the transaction should be recoverd during an IMS emergency or normal restart. If isInquiry=False is coded, isRecover=False cannot be coded.

For IMS Fast Path transactions, isRecover must be True.

If the SPA parameter is specified, the combination of isInquiry=True and isRecover=False must not be coded.

MaxRegions: TMaxRegions = 0

Limits the number of message processing program (MPP) regions that can be concurrently scheduled to process a transaction. When the number of MPP regions is not limited, one transaction can be monopolize all available regions.

If the value is zero, no limit is imposed.

If ApplicationControlMacro.ScheduleType=Serial or TransactionControlMacro.isSerial=True is specified, the MaxRegions parameter must be omitted or zero.

Mode: TMode = MultipleMessage

Specifies that database buffers are to be written to direct access (flushed) upon each request for a new message (SingleMessage) by the processing program, or upon program termination (MultipleMessage). Conversational and WaitForInput transactions must be defined as SingleMessage. SingleMessage is forced for WaitForInput applications.

This operand is not appropriate if a transaction destined for processing in another (remote) system is being specified.

MessageType: TSegment = MultipleSegment

Specifies the type of transaction code (single or multiple segment).

The transaction code can be single segment (SingleSegment), or multiple segment (MultipleSegment). It specifies the time at which an incoming message is considered complete and available to be routed to an application program for subsequent processing.

The meaning of the values in detail:

MultipleSegment: Specifies that the incoming message is one segment in length. It is not eligible for scheduling to an application program until an end-of-message indication is received, or a complete message is created by MFS.

SingleSegment: Specifies that the incoming message is one segment in length. It becomes eligible for scheduling when the terminal operator indicates end-of-segment.

MessageIsResponse: TResponse = NonResponse

MADSON & METCALF, P.C.
ATTORNEYS AT LAW
900 GATEWAY TOWER WEST
15 WEST SOUTH TEMPLE
SALT LAKE CITY, UTAH 84101

Specifies whether or not the communication line from the transaction was entered is to be held until a response is received.

The meaning of the values in detail:

NonResponse: Specifies that, for terminals specifying or defaulting to OPTIONS=TRANRESP, input should not stop after this transaction is entered.

Response: Specifies that, for terminals specifying or defaulting to OPTIONS=TRANRESP, no further messages are to be allowed after this transaction is entered until this transaction sends a response message back to the terminal. Response mode can be forced or negated by individual terminal definition.

MessageClass: TCLass = 1

Specifies the class to which this transaction code is to be assigned.

ParallelLimit: TParallelLimit

Specifies the threshold value to be used when ApplicationControlMacro.ScheduleType=Parallel was specified in the preceding ApplicationControlMacro instruction. An additional region is scheduled whenever the current transaction enqueue count exceeds the ParallelLimit value multiplied by the number of regions currently scheduled for this transaction. If ParallelLimit is not specified, the default value of none is assumed, and IMS allows the transaction to be scheduled in only one region at a time.

ParallelLimit=0 indicates that any input message can cause a new region to be scheduled because the scheduling condition will always be met.

If ParallelLimit=0 is specified, the MaxRegions value should be specified to limit the number of regions that can be scheduled to process a particular transaction.

This operand is not appropriate if a transaction destined for processing in another (remote) system is being specified.

ProcessLimitCount: UnsignedInteger = 65535

Specifies the number of messages (count) of this transaction code a program can process in a single scheduling.

This field specifies the number of messages sent to the application program by the IMS control program for processing without reloading the application program. If 0 is coded, the maximum number of messages sent to the application is one and the application program is reloaded before receiving a subsequent message. The value 65535 means no limit is being set.

This operand is not appropriate if a transaction destined for processing in another (remote) system is being specified.

ProcessLimitSeconds: UnsignedInteger = 65535

MADSON & METCALF, P.C.
ATTORNEYS AT LAW
900 GATEWAY TOWER WEST
15 WEST SOUTH TEMPLE
SALT LAKE CITY, UTAH 84101

Specifies the amount of time (in seconds) allowable of this transaction code to process a single transaction (or message).

If Fast Path is used, the keyword parameter specifies, for a given transaction code, the amount of time (in hundredths of seconds) the program is allowed to process a single transaction message. The time represents real time elapses during transaction processing (not accumulated task time). Real time is used because the input terminal is in response mode and cannot enter another transaction until the response is sent. In this case, the ProcessLimitCount is ignored.

The value 0 is invalid.

This operand is not appropriate if a transaction destined for processing in another (remote) system is being specified.

PriorityNormal: TPriority = 1

Specifies the priority of this transaction when the number of input transactions enqueued and waiting to be processed is less than the PriorityLimitCount value.

PriorityLimit: TPriority = 1

Specifies the priority of this transaction to which it is raised when the number of input transactions, enqueued and waiting to be processed is equal to or greater then the PriorityLimitCount value.

PriorityLimitCount: UnsignedInteger = 65535

Specifies the number that, when compared to the number of input transactions enqueued and waiting to be processed, determines whether the normal or limit priority value is assigned to this transaction.

The value 0 is invalid.

isRouting: Boolean = False

If MSC directed routing is used in a multiple IMS system configuration, specifies whether the application program processing a transaction is informed of the system which originated the transaction.

ScheduleOption: TScheduleOption = 1

Specifies the scheduling option used for other transactions when this transaction cannot be scheduled for internal reasons (database intent or no more space in PSB pool or DMB pool to bring in needed blocks).

The meaning of the values in detail:

1: Schedule only transactions of equal or higher priority in the selected class.

2: Schedule higher priority transactions in the selected class.

3: Schedule any transaction in the selected class.

MADSON & METCALF, P.C.
ATTORNEYS AT LAW
900 GATEWAY TOWER WEST
15 WEST SOUTH TEMPLE
SALT LAKE CITY, UTAH 84101

4: Skip to the next class and attempt to schedule the highest priority transaction in that class.

This operand is not appropriate if a transaction destined for processing in another (remote) system is being specified.

SegmentNumber: UnsignedInteger = 0

Specifies the maximum number of application program output segments that will be allowed into the message queues per Get Unique (GU) call from the application program.

This operand is not appropriate if a transaction destined for processing in another (remote) system is being specified.

SegmentSize: UnsignedInteger = 0

Specifies the maximum number of bytes allowed in any one output segment.

This operand is not appropriate if a transaction destined for processing in another (remote) system is being specified.

isSerial: Boolean = False

Specifies whether the processing of messages for a given transaction is forced to be serial.

If True is specified, ParallelLimit and MaxRegions must equal zero (they can be omitted). If they are not, False is assumed.

This operand is not appropriate if a transaction destined for processing in another (remote) system is being specified.

SPASize: TSpaSize

Defines by inclusion that this transaction is a conversational transaction.

The value specifies the size of the conversational scratchpad area (SPA).

SPAIsTruncated: Boolean = True

Specifies whether the truncated data option is used with the scratchpad area. .

SysIDRemote: TSysID

Specifies in the multiple IMS system configuration, the system identification (SysID) of the remote system (that system on which the application executes).

If the ApplicationControlMacro.SysID parameters are specified, the TransactionControlMacro.SysID parameters don't need to be specified. If both SysID parameters are specified, the ApplicationControlMacro.SysID parameters are ignored.

SysIDLocal: TSysID

Specifies in the multiple IMS system configuration, the system identification (SysID) of the local system (the originating system to which the response are returned).

MADSON & METCALF, P.C.
ATTORNEYS AT LAW
900 GATEWAY TOWER WEST
15 WEST SOUTH TEMPLE
SALT LAKE CITY, UTAH 84101

If the ApplicationControlMacro.SysID parameters are specified, the TransactionControlMacro.SysID parameters don't need to be specified. If both SysID parameters are specified, the ApplicationControlMacro.SysID parameters are ignored.

isWaitForInput: Boolean

Specifies whether (True) or not (False) this is a wait-for-input transaction. A message processing or batch processing application program that processes WaitForInput transactions is scheduled and invoked normally. If the transaction to be processed is defined as WaitForInput, the program is allowed to remain in main storage after it has processed the available input messages. The QC status code (no more messages) is returned to the program if the ProcessLimitCount value is reached.

Mode=SingleMessage will be forced, if isWaitForInput=True is coded.

This operand is not appropriate if a transaction destined for processing in another (remote) system is being specified.

Inherited Attributes

Associations

APPLCTN with ApplicationControlMacro

TResidency enumeration

Attributes

Resident
DynamicOption

TLanguage enumeration

Attributes

Assembler
COBOL
PL/1
PL/I
Pascal

TScheduleType enumeration

Attributes

Serial
Parallel

TProgramType enumeration

Attributes

MADSON & METCALF, P.C.
ATTORNEYS AT LAW
900 GATEWAY TOWER WEST
15 WEST SOUTH TEMPLE
SALT LAKE CITY, UTAH 84101

TeleProcessing
Batch

Boolean enumeration

Attributes

False
True

TMode enumeration

Attributes

MultipleMessage
SingleMessage

MessageType enumeration

Attributes

MULTSEG,NONRESPONSE
SNGLSEG,RESPONSE

String primitive

length: 1..8
valid characters: A through Z, #, $, @, 0 through 9

TSysID primitive

Valid values: 1..2036

TFPBuffer primitive

Valid values: 12..30720

TClass primitive

valid values: 1..255

TPriority primitive

valid values: 0..14

TScheduleOption primitive

valid values: 1..4

TParallelLimit primitive

valid values: 0..32767

MADSON & METCALF, P.C.
ATTORNEYS AT LAW
900 GATEWAY TOWER WEST
15 WEST SOUTH TEMPLE
SALT LAKE CITY, UTAH 84101

TMaxRegions primitive

valid values: 0..255

ProcessLimitType primitive

valid values: 1..65535

UnsignedInteger primitive

valid values: 0..65535

TSpaSize primitive

valid values: 16..32767

TSegment enumeration

Attributes

MultipleSegment

SingleSegment

TResponse enumeration

Attributes

Response

NonResponse

MADSON & METCALF, P.C.
ATTORNEYS AT LAW
900 GATEWAY TOWER WEST
15 WEST SOUTH TEMPLE
SALT LAKE CITY, UTAH 84101

APPENDIX B

Document Type Definition (DTD)

The following is a DTD for IMS transaction definitions according to an embodiment of the invention.

```
<?xml version="1.0" encoding="UTF-8" ?>

<!-- XMI Automatic DTD Generation      -->
<!-- Metamodel: macro                  -->
<!-- ____________________________________________________ -->
<!--                                                      -->
<!-- XMI is the top-level XML element for XMI transfer text -->
<!-- ____________________________________________________ -->

<!ELEMENT XMI (XMI.header, XMI.content?, XMI.difference*,
        XMI.extensions*) >
<!ATTLIST XMI
      xmi.version CDATA #FIXED "1.0"
      timestamp CDATA #IMPLIED
      verified (true | false) #IMPLIED
>

<!-- ____________________________________________________ -->
<!--                                                      -->
<!-- XMI.header contains documentation and identifies the model, -->
<!-- metamodel, and metametamodel                         -->
<!-- ____________________________________________________ -->

<!ELEMENT XMI.header (XMI.documentation?, XMI.model*, XMI.metamodel*,
            XMI.metametamodel*) >

<!-- ____________________________________________________ -->
<!--                                                      -->
<!-- documentation for transfer data                      -->
<!-- ____________________________________________________ -->

<!ELEMENT XMI.documentation (#PCDATA | XMI.owner | XMI.contact |
                  XMI.longDescription | XMI.shortDescription |
                  XMI.exporter | XMI.exporterVersion |
                  XMI.notice)* >

<!ELEMENT XMI.owner ANY >
```

```
<!ELEMENT XMI.contact ANY >
<!ELEMENT XMI.longDescription ANY >
<!ELEMENT XMI.shortDescription ANY >
<!ELEMENT XMI.exporter ANY >
<!ELEMENT XMI.exporterVersion ANY >
<!ELEMENT XMI.exporterID ANY >
<!ELEMENT XMI.notice ANY >

<!-- ____________________________________________________ -->
<!--                                                      -->
<!-- XMI.element.att defines the attributes that each XML element -->
<!-- that corresponds to a metamodel class must have to conform to -->
<!-- the XMI specification.                               -->
<!-- ____________________________________________________ -->

<!ENTITY % XMI.element.att
        'xmi.id ID #IMPLIED xmi.label CDATA #IMPLIED xmi.uuid
         CDATA #IMPLIED ' >

<!-- ____________________________________________________ -->
<!--                                                      -->
<!-- XMI.link.att defines the attributes that each XML element that -->
<!-- corresponds to a metamodel class must have to enable it to -->
<!-- function as a simple XLink as well as refer to model -->
<!-- constructs within the same XMI file.                 -->
<!-- ____________________________________________________ -->

<!ENTITY % XMI.link.att
        'xml:link CDATA #IMPLIED inline (true | false) #IMPLIED
         actuate (show | user) #IMPLIED href CDATA #IMPLIED role
         CDATA #IMPLIED title CDATA #IMPLIED show (embed | replace
         | new) #IMPLIED behavior CDATA #IMPLIED xmi.idref IDREF
         #IMPLIED xmi.uuidref CDATA #IMPLIED' >

<!-- ____________________________________________________ -->
<!--                                                      -->
<!-- XMI.model identifies the model(s) being transferred  -->
<!-- ____________________________________________________ -->

<!ELEMENT XMI.model ANY >
```

MADSON & METCALF, P.C.
ATTORNEYS AT LAW
900 GATEWAY TOWER WEST
15 WEST SOUTH TEMPLE
SALT LAKE CITY, UTAH 84101

```
<!ATTLIST XMI.model
        %XMI.link.att;
        xmi.name    CDATA #REQUIRED
        xmi.version  CDATA #IMPLIED
>

<!-- ____________________________________________________ -->
<!--                                                      -->
<!-- XMI.metamodel identifies the metamodel(s) for the transferred -->
<!-- data                                                 -->
<!-- ____________________________________________________ -->

<!ELEMENT XMI.metamodel ANY >
<!ATTLIST XMI.metamodel
        %XMI.link.att;
        xmi.name    CDATA #REQUIRED
        xmi.version  CDATA #IMPLIED
>

<!-- ____________________________________________________ -->
<!--                                                      -->
<!-- XMI.metametamodel identifies the metametamodel(s) for the -->
<!-- transferred data                                     -->
<!-- ____________________________________________________ -->

<!ELEMENT XMI.metametamodel ANY >
<!ATTLIST XMI.metametamodel
        %XMI.link.att;
        xmi.name    CDATA #REQUIRED
        xmi.version  CDATA #IMPLIED
>

<!-- ____________________________________________________ -->
<!--                                                      -->
<!-- XMI.content is the actual data being transferred     -->
<!-- ____________________________________________________ -->

<!ELEMENT XMI.content ANY >

<!-- ____________________________________________________ -->
<!--                                                      -->
<!-- XMI.extensions contains data to transfer that does not conform -->
<!-- to the metamodel(s) in the header                    -->
<!-- ____________________________________________________ -->

<!ELEMENT XMI.extensions ANY >
```

MADSON & METCALF, P.C.
ATTORNEYS AT LAW
900 GATEWAY TOWER WEST
15 WEST SOUTH TEMPLE
SALT LAKE CITY, UTAH 84101

```
<!ATTLIST XMI.extensions
        xmi.extender CDATA #REQUIRED
>

<!-- ____________________________________________________________ -->
<!--                                                              -->
<!-- extension contains information related to a specific model   -->
<!-- construct that is not defined in the metamodel(s) in the header -->
<!-- ____________________________________________________________ -->

<!ELEMENT XMI.extension ANY >
<!ATTLIST XMI.extension
        %XMI.element.att;
        %XMI.link.att;
        xmi.extender   CDATA #REQUIRED
        xmi.extenderID CDATA #REQUIRED
>

<!-- ____________________________________________________________ -->
<!--                                                              -->
<!-- XMI.difference holds XML elements representing differences to a -->
<!-- base model                                                   -->
<!-- ____________________________________________________________ -->

<!ELEMENT XMI.difference (XMI.difference | XMI.delete | XMI.add |
                XMI.replace)* >
<!ATTLIST XMI.difference
        %XMI.element.att;
        %XMI.link.att;
>


<!-- ____________________________________________________________ -->
<!--                                                              -->
<!-- XMI.delete represents a deletion from a base model           -->
<!-- ____________________________________________________________ -->

<!ELEMENT XMI.delete EMPTY >
<!ATTLIST XMI.delete
        %XMI.element.att;
        %XMI.link.att;
>

<!-- ____________________________________________________________ -->
<!--                                                              -->
<!-- XMI.add represents an addition to a base model               -->
```

MADSON & METCALF, P.C.
ATTORNEYS AT LAW
900 GATEWAY TOWER WEST
15 WEST SOUTH TEMPLE
SALT LAKE CITY, UTAH 84101

```
<!-- ____________________________________________________ -->

<!ELEMENT XMI.add ANY >
<!ATTLIST XMI.add
        %XMI.element.att;
        %XMI.link.att;
        xmi.position CDATA "-1"
>

<!-- ____________________________________________________ -->
<!--                                                      -->
<!-- XMI.replace represents the replacement of a model construct -->
<!-- with another model construct in a base model         -->
<!-- ____________________________________________________ -->

<!ELEMENT XMI.replace ANY >
<!ATTLIST XMI.replace
        %XMI.element.att;
        %XMI.link.att;
        xmi.position CDATA "-1"
>

<!-- ____________________________________________________ -->
<!--                                                      -->
<!-- XMI.reference may be used to refer to data types not defined in -->
<!-- the metamodel                                        -->
<!-- ____________________________________________________ -->

<!ELEMENT XMI.reference ANY >
<!ATTLIST XMI.reference
        %XMI.link.att;
>

<!-- ____________________________________________________ -->
<!--                                                      -->
<!-- This section contains the declaration of XML elements -->
<!-- representing data types                              -->
<!-- ____________________________________________________ -->

<!ELEMENT XMI.TypeDefinitions ANY >

<!ELEMENT XMI.field ANY >

<!ELEMENT XMI.seqItem ANY >

<!ELEMENT XMI.octetStream (#PCDATA) >
```

MADSON & METCALF, P.C.
ATTORNEYS AT LAW
900 GATEWAY TOWER WEST
15 WEST SOUTH TEMPLE
SALT LAKE CITY, UTAH 84101

```
<!ELEMENT XMI.unionDiscrim ANY >

<!ELEMENT XMI.enum EMPTY >
<!ATTLIST XMI.enum
        xmi.value CDATA #REQUIRED
>

<!ELEMENT XMI.any ANY >
<!ATTLIST XMI.any
        %XMI.link.att;
        xmi.type CDATA #IMPLIED
        xmi.name CDATA #IMPLIED
>

<!ELEMENT XMI.CorbaTypeCode (XMI.CorbaTcAlias | XMI.CorbaTcStruct |
                        XMI.CorbaTcSequence | XMI.CorbaTcArray |
                        XMI.CorbaTcEnum | XMI.CorbaTcUnion |
                        XMI.CorbaTcExcept | XMI.CorbaTcString |
                        XMI.CorbaTcWstring | XMI.CorbaTcShort |
                        XMI.CorbaTcLong | XMI.CorbaTcUshort |
                        XMI.CorbaTcUlong | XMI.CorbaTcFloat |
                        XMI.CorbaTcDouble | XMI.CorbaTcBoolean |
                        XMI.CorbaTcChar | XMI.CorbaTcWchar |
                        XMI.CorbaTcOctet | XMI.CorbaTcAny |
                        XMI.CorbaTcTypeCode | XMI.CorbaTcPrincipal |
                        XMI.CorbaTcNull | XMI.CorbaTcVoid |
                        XMI.CorbaTcLongLong |
                        XMI.CorbaTcLongDouble) >
<!ATTLIST XMI.CorbaTypeCode
        %XMI.element.att;
>

<!ELEMENT XMI.CorbaTcAlias (XMI.CorbaTypeCode) >
<!ATTLIST XMI.CorbaTcAlias
        xmi.tcName CDATA #REQUIRED
        xmi.tcId   CDATA #IMPLIED
>

<!ELEMENT XMI.CorbaTcStruct (XMI.CorbaTcField)* >
<!ATTLIST XMI.CorbaTcStruct
        xmi.tcName CDATA #REQUIRED
        xmi.tcId   CDATA #IMPLIED
>

<!ELEMENT XMI.CorbaTcField (XMI.CorbaTypeCode) >
```

MADSON & METCALF, P.C.
ATTORNEYS AT LAW
900 GATEWAY TOWER WEST
15 WEST SOUTH TEMPLE
SALT LAKE CITY, UTAH 84101

```
<!ATTLIST XMI.CorbaTcField
        xmi.tcName CDATA #REQUIRED
>

<!ELEMENT XMI.CorbaTcSequence (XMI.CorbaTypeCode |
                XMI.CorbaRecursiveType) >
<!ATTLIST XMI.CorbaTcSequence
        xmi.tcLength CDATA #REQUIRED
>

<!ELEMENT XMI.CorbaRecursiveType EMPTY >
<!ATTLIST XMI.CorbaRecursiveType
        xmi.offset CDATA #REQUIRED
>

<!ELEMENT XMI.CorbaTcArray (XMI.CorbaTypeCode) >
<!ATTLIST XMI.CorbaTcArray
        xmi.tcLength CDATA #REQUIRED
>

<!ELEMENT XMI.CorbaTcObjRef EMPTY >
<!ATTLIST XMI.CorbaTcObjRef
        xmi.tcName CDATA #REQUIRED
        xmi.tcId   CDATA #IMPLIED
>

<!ELEMENT XMI.CorbaTcEnum (XMI.CorbaTcEnumLabel) >
<!ATTLIST XMI.CorbaTcEnum
        xmi.tcName CDATA #REQUIRED
        xmi.tcId   CDATA #IMPLIED
>

<!ELEMENT XMI.CorbaTcEnumLabel EMPTY >
<!ATTLIST XMI.CorbaTcEnumLabel
        xmi.tcName CDATA #REQUIRED
>

<!ELEMENT XMI.CorbaTcUnionMbr (XMI.CorbaTypeCode, XMI.any) >
<!ATTLIST XMI.CorbaTcUnionMbr
        xmi.tcName CDATA #REQUIRED
>

<!ELEMENT XMI.CorbaTcUnion (XMI.CorbaTypeCode, XMI.CorbaTcUnionMbr*) >
<!ATTLIST XMI.CorbaTcUnion
        xmi.tcName CDATA #REQUIRED
        xmi.tcId   CDATA #IMPLIED>
```

MADSON & METCALF, P.C.
ATTORNEYS AT LAW
900 GATEWAY TOWER WEST
15 WEST SOUTH TEMPLE
SALT LAKE CITY, UTAH 84101

```
<!ELEMENT XMI.CorbaTcExcept (XMI.CorbaTcField)* >
<!ATTLIST XMI.CorbaTcExcept
        xmi.tcName CDATA #REQUIRED
        xmi.tcId   CDATA #IMPLIED
>

<!ELEMENT XMI.CorbaTcString EMPTY >
<!ATTLIST XMI.CorbaTcString
        xmi.tcLength CDATA #REQUIRED
>

<!ELEMENT XMI.CorbaTcWstring EMPTY >
<!ATTLIST XMI.CorbaTcWstring
        xmi.tcLength CDATA #REQUIRED
>

<!ELEMENT XMI.CorbaTcFixed EMPTY >
<!ATTLIST XMI.CorbaTcFixed
        xmi.tcDigits CDATA #REQUIRED
        xmi.tcScale  CDATA #REQUIRED
>

<!ELEMENT XMI.CorbaTcShort EMPTY >

<!ELEMENT XMI.CorbaTcLong EMPTY >

<!ELEMENT XMI.CorbaTcUshort EMPTY >

<!ELEMENT XMI.CorbaTcUlong EMPTY >

<!ELEMENT XMI.CorbaTcFloat EMPTY >

<!ELEMENT XMI.CorbaTcDouble EMPTY >

<!ELEMENT XMI.CorbaTcBoolean EMPTY >

<!ELEMENT XMI.CorbaTcChar EMPTY >

<!ELEMENT XMI.CorbaTcWchar EMPTY >

<!ELEMENT XMI.CorbaTcOctet EMPTY >

<!ELEMENT XMI.CorbaTcAny EMPTY >

<!ELEMENT XMI.CorbaTcTypeCode EMPTY >
```

MADSON & METCALF, P.C.
ATTORNEYS AT LAW
900 GATEWAY TOWER WEST
15 WEST SOUTH TEMPLE
SALT LAKE CITY, UTAH 84101

```
<!ELEMENT XMI.CorbaTcPrincipal EMPTY >
<!ELEMENT XMI.CorbaTcNull EMPTY >
<!ELEMENT XMI.CorbaTcVoid EMPTY >
<!ELEMENT XMI.CorbaTcLongLong EMPTY >
<!ELEMENT XMI.CorbaTcLongDouble EMPTY >

<!-- ____________________________________________________ -->
<!--                                                      -->
<!-- METAMODEL:  macro                                    -->
<!-- ____________________________________________________ -->


<!-- ____________________________________________________ -->
<!--                                                      -->
<!-- METAMODEL CLASS:  ApplicationControlMacro            -->
<!-- ____________________________________________________ -->

<!ELEMENT ApplicationControlMacro.ResidencyOption EMPTY >
<!ATTLIST ApplicationControlMacro.ResidencyOption
        xmi.value ( Resident | DynamicOption ) #REQUIRED
>

<!ELEMENT ApplicationControlMacro.isFPath EMPTY >
<!ATTLIST ApplicationControlMacro.isFPath
        xmi.value ( False | True ) #REQUIRED
>

<!ELEMENT ApplicationControlMacro.FPathBufferSize (#PCDATA |
                                XMI.reference)* >

<!ELEMENT ApplicationControlMacro.GeneratedPSBName (#PCDATA |
                                XMI.reference)* >

<!ELEMENT ApplicationControlMacro.GeneratedPSBLanguage EMPTY >
<!ATTLIST ApplicationControlMacro.GeneratedPSBLanguage
        xmi.value ( Assembler | COBOL | PL1 | PLI | Pascal ) #REQUIRED
>

<!ELEMENT ApplicationControlMacro.ProgramType EMPTY >
<!ATTLIST ApplicationControlMacro.ProgramType
        xmi.value ( TeleProcessing | Batch ) #REQUIRED
```

MADSON & METCALF, P.C.
ATTORNEYS AT LAW
900 GATEWAY TOWER WEST
15 WEST SOUTH TEMPLE
SALT LAKE CITY, UTAH 84101

```
>

<!ELEMENT ApplicationControlMacro.ProgramClass (#PCDATA |
                                XMI.reference)* >

<!ELEMENT ApplicationControlMacro.PSBName (#PCDATA | XMI.reference)* >

<!ELEMENT ApplicationControlMacro.ScheduleType EMPTY >
<!ATTLIST ApplicationControlMacro.ScheduleType
        xmi.value ( Serial | Parallel ) #REQUIRED
>

<!ELEMENT ApplicationControlMacro.SysIDRemote (#PCDATA |
                                XMI.reference)* >

<!ELEMENT ApplicationControlMacro.SysIDLocal (#PCDATA |
                                XMI.reference)* >

<!ELEMENT ApplicationControlMacro.TRANSACT (TransactionControlMacro)* >

<!ELEMENT ApplicationControlMacro (ApplicationControlMacro.ResidencyOption?,
                        ApplicationControlMacro.isFPath?,
                        ApplicationControlMacro.FPathBufferSize?,
                        ApplicationControlMacro.GeneratedPSBName?,
                        ApplicationControlMacro.GeneratedPSBLanguage?,
                        ApplicationControlMacro.ProgramType?,
                        ApplicationControlMacro.ProgramClass?,
                        ApplicationControlMacro.PSBName?,
                        ApplicationControlMacro.ScheduleType?,
                        ApplicationControlMacro.SysIDRemote?,
                        ApplicationControlMacro.SysIDLocal?,
                        XMI.extension*,
                        ApplicationControlMacro.TRANSACT*)? >
<!ATTLIST ApplicationControlMacro
        %XMI.element.att;
        %XMI.link.att;
>

<!-- ____________________________________________________________ -->
<!--                                                              -->
<!-- METAMODEL CLASS:  TransactionControlMacro                    -->
<!-- ____________________________________________________________ -->

<!ELEMENT TransactionControlMacro.TransactionCode (#PCDATA |
                                XMI.reference)* >
```

MADSON & METCALF, P.C.
ATTORNEYS AT LAW
900 GATEWAY TOWER WEST
15 WEST SOUTH TEMPLE
SALT LAKE CITY, UTAH 84101

```
<!ELEMENT TransactionControlMacro.isDCLogWriteAhead EMPTY >
<!ATTLIST TransactionControlMacro.isDCLogWriteAhead
        xmi.value ( False | True ) #REQUIRED
>

<!ELEMENT TransactionControlMacro.toUpperCase EMPTY >
<!ATTLIST TransactionControlMacro.toUpperCase
        xmi.value ( False | True ) #REQUIRED
>

<!ELEMENT TransactionControlMacro.EditRoutine (#PCDATA |
                                XMI.reference)* >

<!ELEMENT TransactionControlMacro.isFPath EMPTY >
<!ATTLIST TransactionControlMacro.isFPath
        xmi.value ( False | True ) #REQUIRED
>

<!ELEMENT TransactionControlMacro.FPathBufferSize (#PCDATA |
                                XMI.reference)* >

<!ELEMENT TransactionControlMacro.isInquiry EMPTY >
<!ATTLIST TransactionControlMacro.isInquiry
        xmi.value ( False | True ) #REQUIRED
>

<!ELEMENT TransactionControlMacro.isRecover EMPTY >
<!ATTLIST TransactionControlMacro.isRecover
        xmi.value ( False | True ) #REQUIRED
>

<!ELEMENT TransactionControlMacro.MaxRegions (#PCDATA |
                                XMI.reference)* >

<!ELEMENT TransactionControlMacro.Mode EMPTY >
<!ATTLIST TransactionControlMacro.Mode
        xmi.value ( MultipleMessage | SingleMessage ) #REQUIRED
>

<!ELEMENT TransactionControlMacro.MessageType EMPTY >
<!ATTLIST TransactionControlMacro.MessageType
        xmi.value ( MultipleSegment | SingleSegment ) #REQUIRED
>

<!ELEMENT TransactionControlMacro.MessageIsResponse EMPTY >
```

MADSON & METCALF, P.C.
ATTORNEYS AT LAW
900 GATEWAY TOWER WEST
15 WEST SOUTH TEMPLE
SALT LAKE CITY, UTAH 84101

```
<!ATTLIST TransactionControlMacro.MessageIsResponse
        xmi.value ( Response | NonResponse ) #REQUIRED
>

<!ELEMENT TransactionControlMacro.MessageClass (#PCDATA |
                                XMI.reference)* >

<!ELEMENT TransactionControlMacro.ParallelLimit (#PCDATA |
                                XMI.reference)* >

<!ELEMENT TransactionControlMacro.ProcessLimitCount (#PCDATA |
                                XMI.reference)* >

<!ELEMENT TransactionControlMacro.ProcessLimitSeconds (#PCDATA |
                                XMI.reference)* >

<!ELEMENT TransactionControlMacro.PriorityNormal (#PCDATA |
                                XMI.reference)* >

<!ELEMENT TransactionControlMacro.PriorityLimit (#PCDATA |
                                XMI.reference)* >

<!ELEMENT TransactionControlMacro.PriorityLimitCount (#PCDATA |
                                XMI.reference)* >


<!ELEMENT TransactionControlMacro.isRouting EMPTY >
<!ATTLIST TransactionControlMacro.isRouting
        xmi.value ( False | True ) #REQUIRED
>

<!ELEMENT TransactionControlMacro.ScheduleOption (#PCDATA |
                                XMI.reference)* >

<!ELEMENT TransactionControlMacro.SegmentNumber (#PCDATA |
                                XMI.reference)* >

<!ELEMENT TransactionControlMacro.SegmentSize (#PCDATA |
                                XMI.reference)* >

<!ELEMENT TransactionControlMacro.isSerial EMPTY >
<!ATTLIST TransactionControlMacro.isSerial
        xmi.value ( False | True ) #REQUIRED
>

<!ELEMENT TransactionControlMacro.SPASize (#PCDATA | XMI.reference)* >
```

MADSON & METCALF, P.C.
ATTORNEYS AT LAW
900 GATEWAY TOWER WEST
15 WEST SOUTH TEMPLE
SALT LAKE CITY, UTAH 84101

```
<!ELEMENT TransactionControlMacro.SPAIsTruncated EMPTY >
<!ATTLIST TransactionControlMacro.SPAIsTruncated
        xmi.value ( False | True ) #REQUIRED
>

<!ELEMENT TransactionControlMacro.SysIDRemote (#PCDATA |
                                XMI.reference)* >

<!ELEMENT TransactionControlMacro.SysIDLocal (#PCDATA |
                                XMI.reference)* >

<!ELEMENT TransactionControlMacro.isWaitForInput EMPTY >
<!ATTLIST TransactionControlMacro.isWaitForInput
        xmi.value ( False | True ) #REQUIRED
>

<!ELEMENT TransactionControlMacro.APPLCTN (ApplicationControlMacro)? >

<!ELEMENT TransactionControlMacro (TransactionControlMacro.TransactionCode?,
                        TransactionControlMacro.isDCLogWriteAhead?,
                        TransactionControlMacro.toUpperCase?,
                        TransactionControlMacro.EditRoutine?,
                        TransactionControlMacro.isFPath?,
                        TransactionControlMacro.FPathBufferSize?,
                        TransactionControlMacro.isInquiry?,
                        TransactionControlMacro.isRecover?,
                        TransactionControlMacro.MaxRegions?,
                        TransactionControlMacro.Mode?,
                        TransactionControlMacro.MessageType?,
                        TransactionControlMacro.MessageIsResponse?,
                        TransactionControlMacro.MessageClass?,
                        TransactionControlMacro.ParallelLimit?,
                        TransactionControlMacro.ProcessLimitCount?,
                        TransactionControlMacro.ProcessLimitSeconds?,
                        TransactionControlMacro.PriorityNormal?,
                        TransactionControlMacro.PriorityLimit?,
                        TransactionControlMacro.PriorityLimitCount?,
                        TransactionControlMacro.isRouting?,
                        TransactionControlMacro.ScheduleOption?,
                        TransactionControlMacro.SegmentNumber?,
                        TransactionControlMacro.SegmentSize?,
                        TransactionControlMacro.isSerial?,
                        TransactionControlMacro.SPASize?,
                        TransactionControlMacro.SPAIsTruncated?,
                        TransactionControlMacro.SysIDRemote?,
```

MADSON & METCALF, P.C.
ATTORNEYS AT LAW
900 GATEWAY TOWER WEST
15 WEST SOUTH TEMPLE
SALT LAKE CITY, UTAH 84101

```
                    TransactionControlMacro.SysIDLocal?,
                    TransactionControlMacro.isWaitForInput?,
                    XMI.extension*,
                    TransactionControlMacro.APPLCTN?)? >
<!ATTLIST TransactionControlMacro
        %XMI.element.att;
        %XMI.link.att;
>

<!ELEMENT macro ((ApplicationControlMacro | TransactionControlMacro)*) >
<!ATTLIST macro
        %XMI.element.att;
        %XMI.link.att;
>
```

MADSON & METCALF, P.C.
ATTORNEYS AT LAW
900 GATEWAY TOWER WEST
15 WEST SOUTH TEMPLE
SALT LAKE CITY, UTAH 84101

Claims

1. A method for communicating with an Information Management System (IMS) using eXtensible Markup Language (XML) documents, the method comprising:

receiving a document comprising an IMS transaction definition encoded in XML;

obtaining a Document Type Definition (DTD) specifying rules for decoding the IMS transaction definition;

parsing the XML document using the DTD to decode the IMS transaction definition; and

providing the decoded IMS transaction definition to the IMS.

2. The method of claim 1, wherein the IMS transaction definition comprises an APPLCTN macro.

3. The method of claim 1, wherein the IMS transaction definition comprises a TRANSACT macro.

4. The method of claim 1, wherein the DTD comprises an XML Metadata Interchange (XMI) DTD.

MADSON & METCALF, P.C.
ATTORNEYS AT LAW
900 GATEWAY TOWER WEST
15 WEST SOUTH TEMPLE
SALT LAKE CITY, UTAH 84101

5. The method of claim 1, wherein the receiving step comprises receiving the document at an IMS gateway.

6. The method of claim 1, wherein the obtaining step comprises:

modeling an IMS transaction definition in a Universal Modeling Language (UML) to produce a UML object model;

processing the UML object model using an XML Metadata Interchange (XMI) utility to create the DTD.

7. The method of claim 1, further comprising:

obtaining an IMS transaction definition;

obtaining a Document Type Definition (DTD) specifying rules for encoding the IMS transaction definition; and

parsing the IMS transaction definition with the DTD to encode the IMS transaction definition in an XML document.

8. A system for communicating with an Information Management System (IMS) using eXtensible Markup Language (XML) documents, the system comprising:

a document reception module configured to receive a document comprising an IMS transaction definition encoded in XML;

MADSON & METCALF, P.C.
ATTORNEYS AT LAW
900 GATEWAY TOWER WEST
15 WEST SOUTH TEMPLE
SALT LAKE CITY, UTAH 84101

a parser configured to obtain a Document Type Definition (DTD) specifying rules for decoding the IMS transaction definition and further configured to parse the XML document using the DTD to decode the IMS transaction definition; and

an IMS interface module configured to provide the decoded IMS transaction definition to the IMS.

9. The system of claim 8, wherein the IMS transaction definition comprises an APPLCTN macro.

10. The system of claim 8, wherein the IMS transaction definition comprises a TRANSACT macro.

11. The system of claim 8, wherein the DTD comprises an XML Metadata Interchange (XMI) DTD.

12. The system of claim 8, wherein the document reception module comprises an IMS gateway.

13. The system of claim 8, further comprising:

MADSON & METCALF, P.C.
ATTORNEYS AT LAW
900 GATEWAY TOWER WEST
15 WEST SOUTH TEMPLE
SALT LAKE CITY, UTAH 84101

a modeling tool configured to model an IMS transaction definition in a Universal Modeling Language (UML) to produce a UML object model; and

an XML Metadata Interchange (XMI) utility configured to process the UML object model to create the DTD.

14. The system of claim 8, wherein the parser is further configured to obtain a DTD specifying rules for encoding an IMS transaction definition as an XML document and to parse the IMS transaction definition using the DTD.

15. An article of manufacture comprising a program storage medium readable by a processor and embodying one or more instructions executable by the processor to perform a method for communicating with an Information Management System (IMS) using eXtensible Markup Language (XML) documents, the method comprising:

receiving a document comprising an IMS transaction definition encoded in XML;

obtaining a Document Type Definition (DTD) specifying rules for decoding the IMS transaction definition;

MADSON & METCALF, P.C.
ATTORNEYS AT LAW
900 GATEWAY TOWER WEST
15 WEST SOUTH TEMPLE
SALT LAKE CITY, UTAH 84101

parsing the XML document using the DTD to decode the IMS transaction definition; and

providing the decoded IMS transaction definition to the IMS.

16. The article of manufacture of claim 15, wherein the IMS transaction definition comprises an APPLCTN macro.

17. The article of manufacture of claim 15, wherein the IMS transaction definition comprises a TRANSACT macro.

18. The article of manufacture of claim 15, wherein the DTD comprises an XML Metadata Interchange (XMI) DTD.

19. The article of manufacture of claim 15, wherein the receiving step comprises receiving the document at an IMS gateway.

20. The article of manufacture of claim 15, wherein the obtaining step comprises:

MADSON & METCALF, P.C.
ATTORNEYS AT LAW
900 GATEWAY TOWER WEST
15 WEST SOUTH TEMPLE
SALT LAKE CITY, UTAH 84101

modeling the IMS transaction definition in a Universal Modeling Language (UML) to produce a UML object model;

processing the UML object model using an XML Metadata Interchange (XMI)utility to create the DTD.

21. The article of manufacture of claim 15, the method further comprising:

obtaining an IMS transaction definition;

obtaining a Document Type Definition (DTD) specifying rules for encoding the IMS transaction definition; and

parsing the IMS transaction definition with the DTD to encode the IMS transaction definition in an XML document.

MADSON & METCALF, P.C.
ATTORNEYS AT LAW
900 GATEWAY TOWER WEST
15 WEST SOUTH TEMPLE
SALT LAKE CITY, UTAH 84101

REPRESENTING IMS TRANSACTION DEFINITIONS AS XML DOCUMENTS

ABSTRACT OF THE DISCLOSURE

A method for communicating with an Information Management System (IMS) using eXtensible Markup Language (XML) documents includes the steps of receiving a document comprising an IMS transaction definition encoded in XML; obtaining a Document Type Definition (DTD) specifying rules for decoding the IMS transaction definition; parsing the XML document using the DTD to decode the IMS transaction definition; and providing the decoded IMS transaction definition to the IMS.

A system for communicating with an IMS using XML documents includes a document reception module configured to receive a document comprising an IMS transaction definition encoded in XML; a parser configured to obtain a Document Type Definition (DTD) specifying rules for decoding the IMS transaction definition and further configured to parse the XML document using the DTD to decode the IMS transaction definition; and an IMS interface module configured to provide the decoded IMS transaction definition to the IMS.

MADSON & METCALF, P.C.
ATTORNEYS AT LAW
900 GATEWAY TOWER WEST
15 WEST SOUTH TEMPLE
SALT LAKE CITY, UTAH 84101

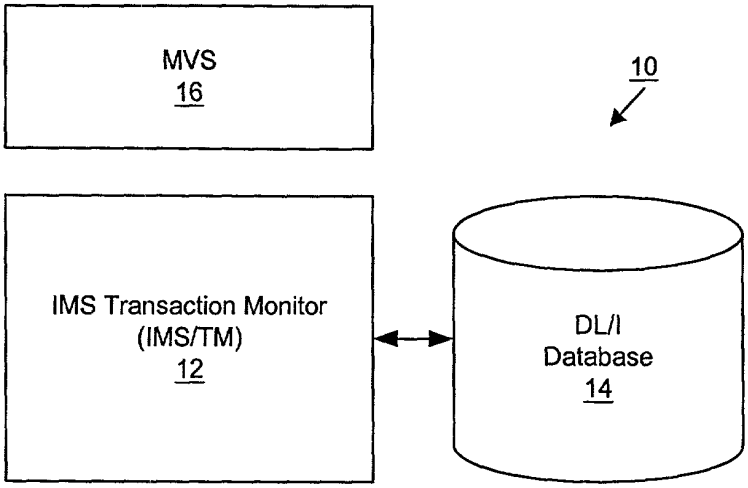

Fig. 1
(prior art)

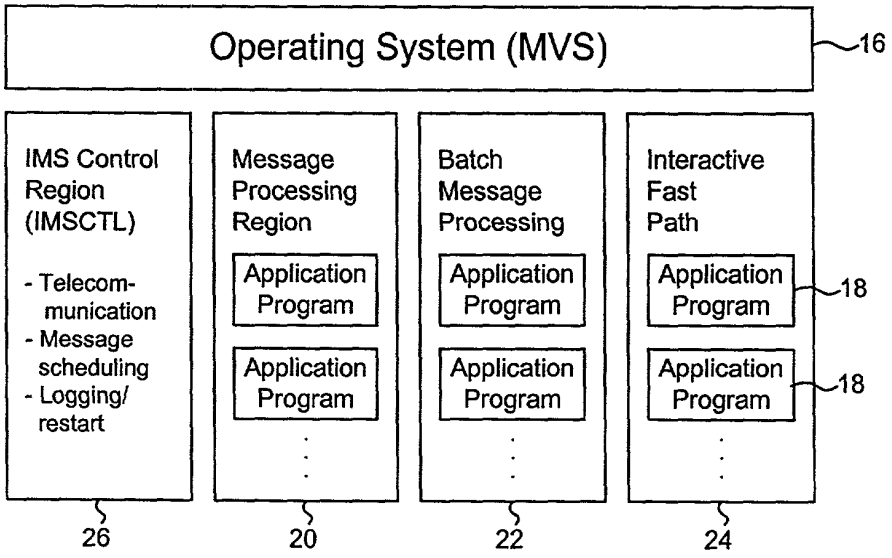

Fig. 2
(prior art)

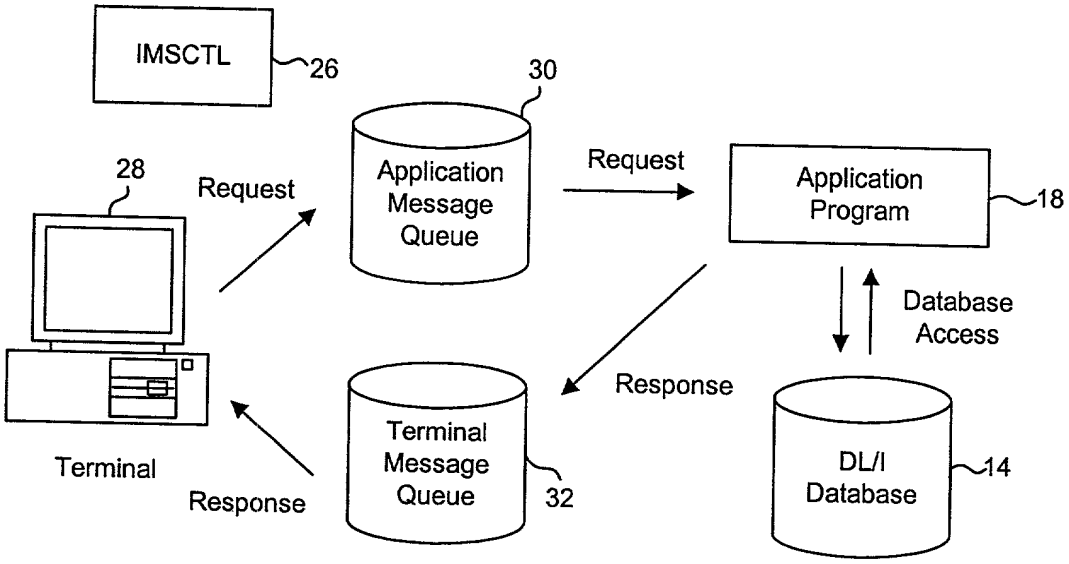

Fig. 3
(prior art)

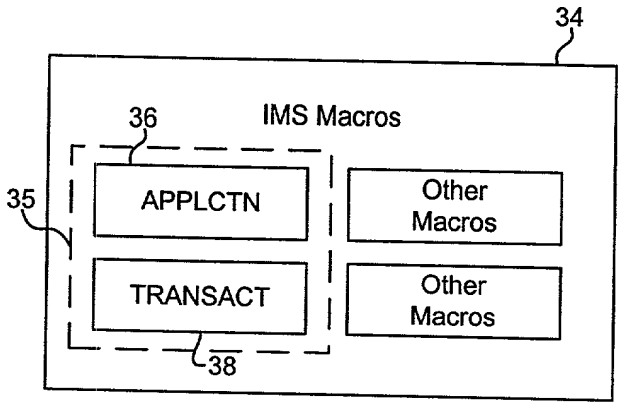

Fig. 4
(prior art)

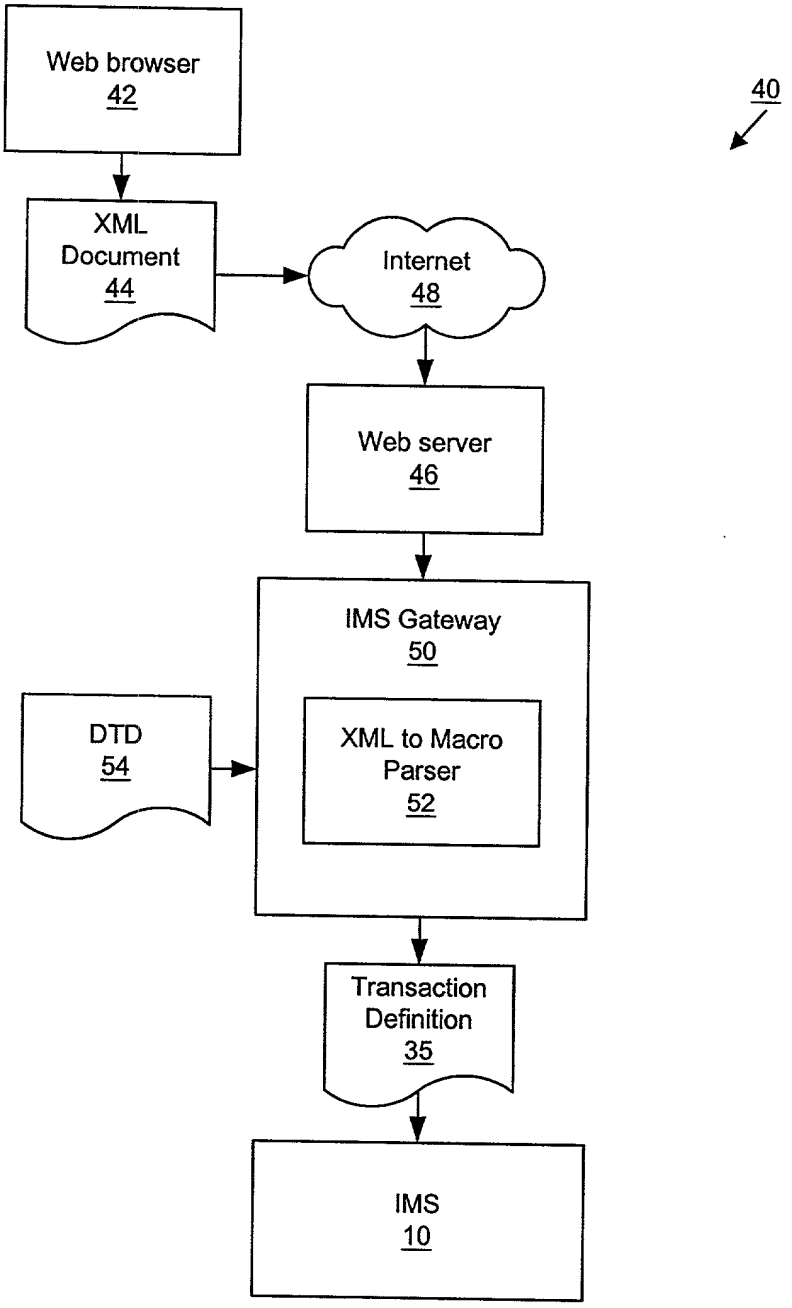
Web browser
42
40
XML
Document
44
Internet
48
Web server
46
IMS Gateway
50
DTD
54
XML to Macro
Parser
52
Transaction
Definition
35
IMS
10

Fig. 5

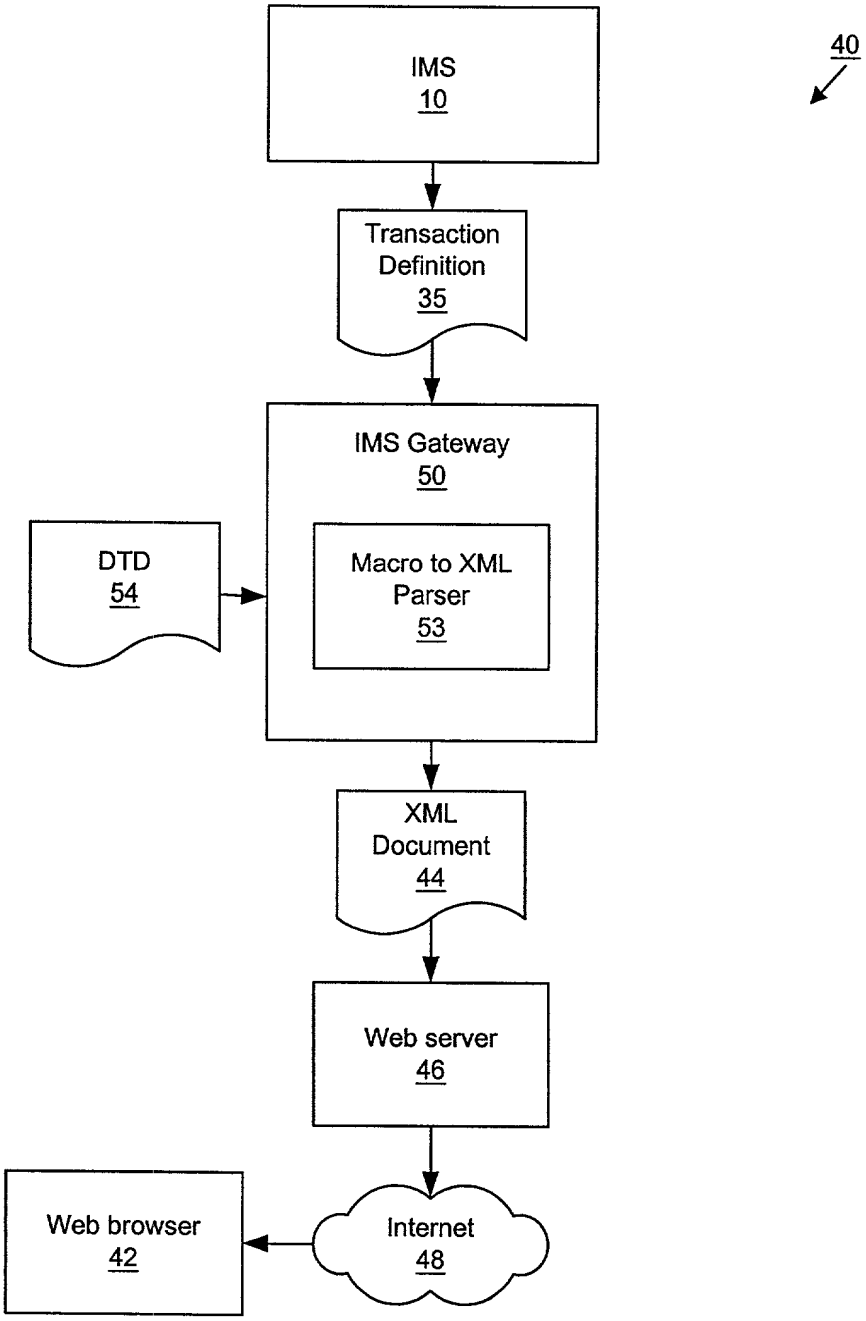
IMS
10
40
Transaction
Definition
35
IMS Gateway
50
DTD
54
Macro to XML
Parser
53
XML
Document
44
Web server
46
Web browser
42
Internet
48

Fig. 6

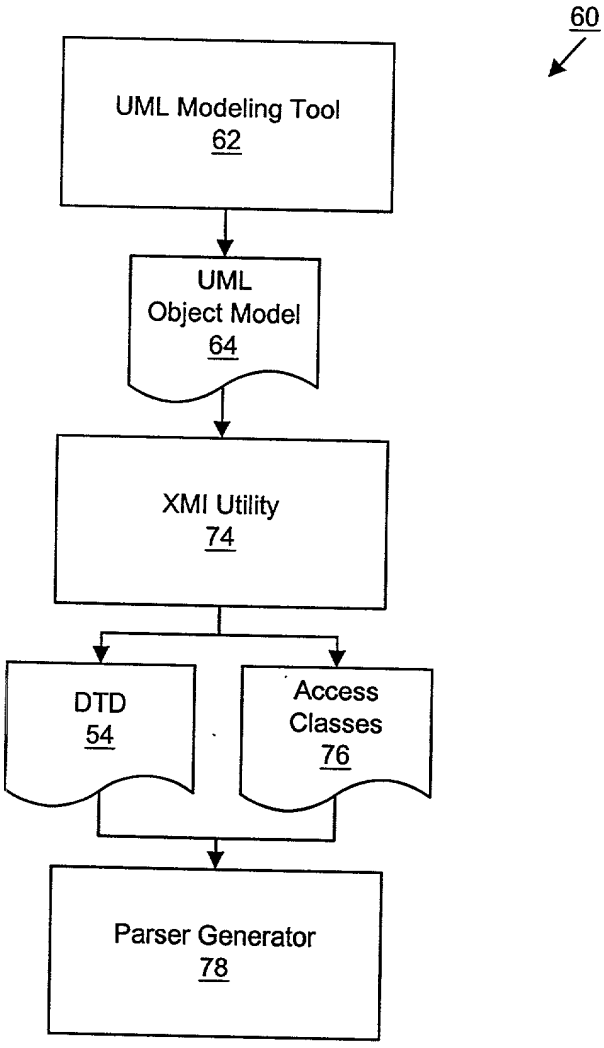
60
UML Modeling Tool
62
UML
Object Model
64
XMI Utility
74
DTD
54
Access
Classes
76
Parser Generator
78

Fig. 7

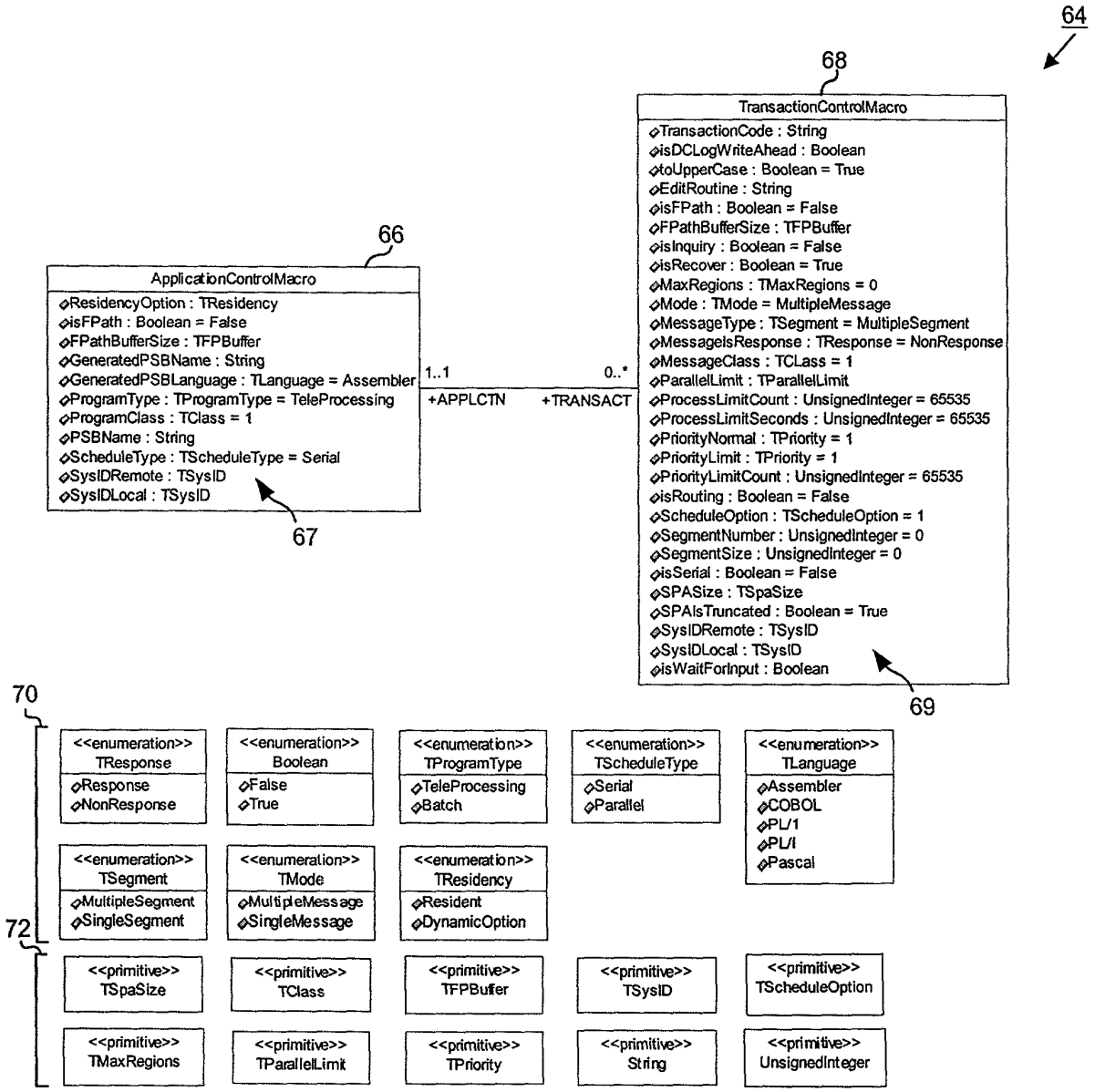
64
68
TransactionControlMacro
◇TransactionCode : String
◇isDCLogWriteAhead : Boolean
◇toUpperCase : Boolean = True
◇EditRoutine : String
◇isFPath : Boolean = False
◇FPathBufferSize : TFPBuffer
◇isInquiry : Boolean = False
◇isRecover : Boolean = True
◇MaxRegions : TMaxRegions = 0
◇Mode : TMode = MultipleMessage
◇MessageType : TSegment = MultipleSegment
◇MessageIsResponse : TResponse = NonResponse
◇MessageClass : TCLass = 1
◇ParallelLimit : TParallelLimit
◇ProcessLimitCount : UnsignedInteger = 65535
◇ProcessLimitSeconds : UnsignedInteger = 65535
◇PriorityNormal : TPriority = 1
◇PriorityLimit : TPriority = 1
◇PriorityLimitCount : UnsignedInteger = 65535
◇isRouting : Boolean = False
◇ScheduleOption : TScheduleOption = 1
◇SegmentNumber : UnsignedInteger = 0
◇SegmentSize : UnsignedInteger = 0
◇isSerial : Boolean = False
◇SPASize : TSpaSize
◇SPAIsTruncated : Boolean = True
◇SysIDRemote : TSysID
◇SysIDLocal : TSysID
◇isWaitForInput : Boolean
69
66
ApplicationControlMacro
◇ResidencyOption : TResidency
◇isFPath : Boolean = False
◇FPathBufferSize : TFPBuffer
◇GeneratedPSBName : String
◇GeneratedPSBLanguage : TLanguage = Assembler
◇ProgramType : TProgramType = TeleProcessing
◇ProgramClass : TClass = 1
◇PSBName : String
◇ScheduleType : TScheduleType = Serial
◇SysIDRemote : TSysID
◇SysIDLocal : TSysID
67
1..1
0..*
+APPLCTN
+TRANSACT
70
72
<<enumeration>> TResponse
◇Response
◇NonResponse
<<enumeration>> Boolean
◇False
◇True
<<enumeration>> TProgramType
◇TeleProcessing
◇Batch
<<enumeration>> TScheduleType
◇Serial
◇Parallel
<<enumeration>> TLanguage
◇Assembler
◇COBOL
◇PL/1
◇PL/I
◇Pascal
<<enumeration>> TSegment
◇MultipleSegment
◇SingleSegment
<<enumeration>> TMode
◇MultipleMessage
◇SingleMessage
<<enumeration>> TResidency
◇Resident
◇DynamicOption
<<primitive>> TSpaSize
<<primitive>> TClass
<<primitive>> TFPBuffer
<<primitive>> TSysID
<<primitive>> TScheduleOption
<<primitive>> TMaxRegions
<<primitive>> TParallelLimit
<<primitive>> TPriority
<<primitive>> String
<<primitive>> UnsignedInteger

Fig. 8

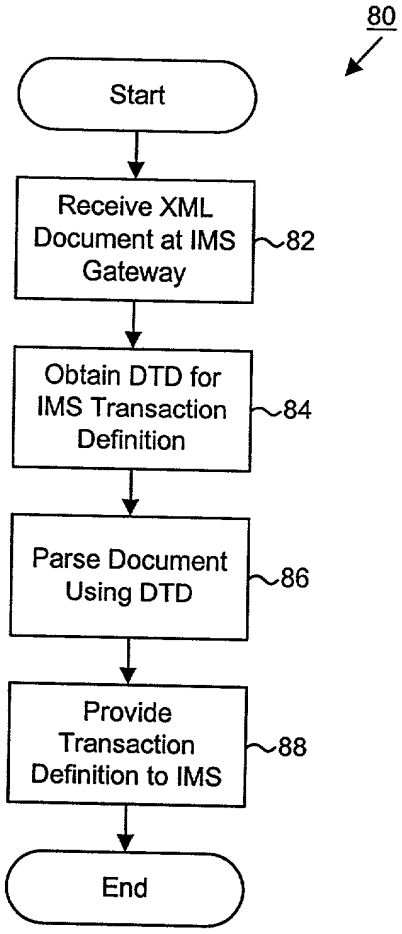
80
Start
Receive XML Document at IMS Gateway
82
Obtain DTD for IMS Transaction Definition
84
Parse Document Using DTD
86
Provide Transaction Definition to IMS
88
End

Fig. 9

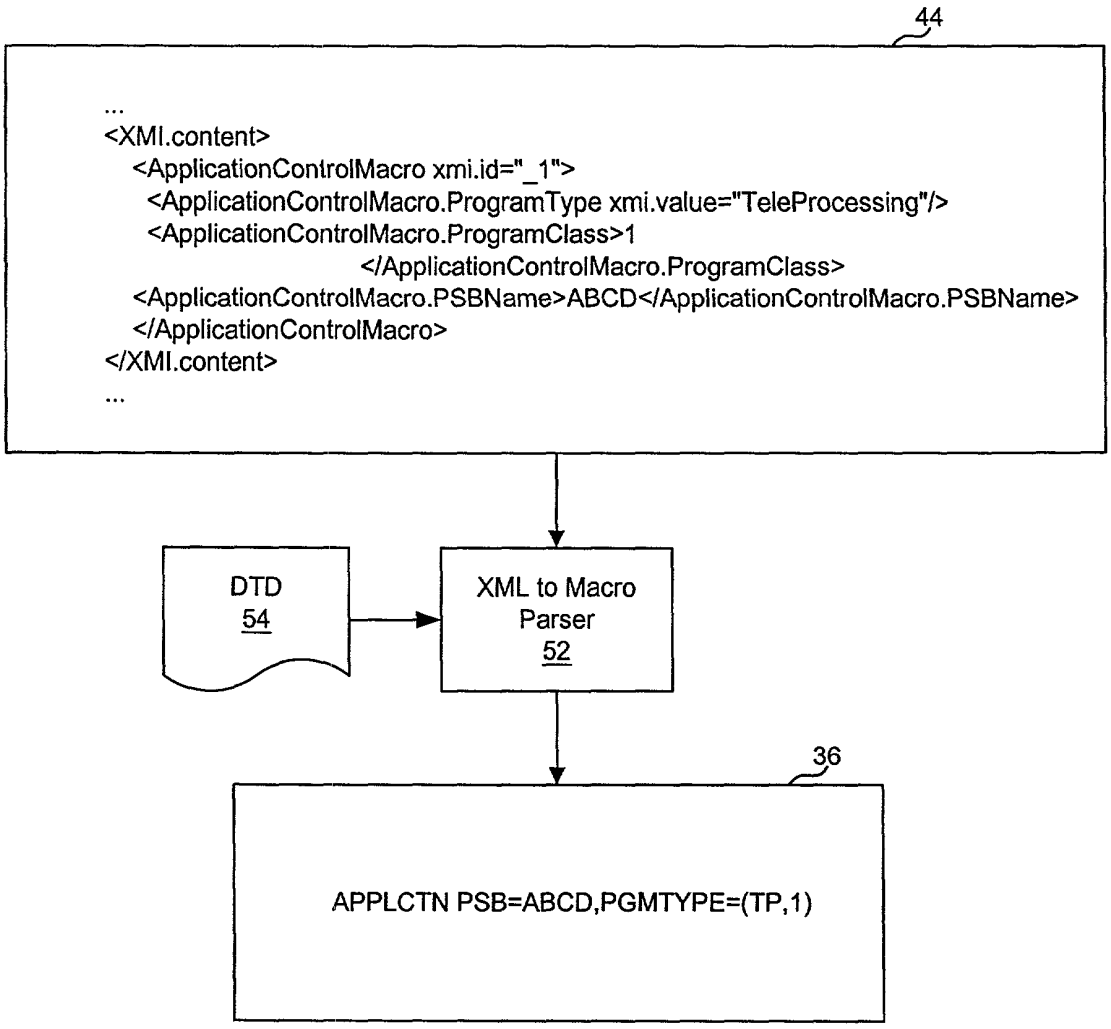
44
...
<XMI.content>
<ApplicationControlMacro xmi.id="_1">
<ApplicationControlMacro.ProgramType xmi.value="TeleProcessing"/>
<ApplicationControlMacro.ProgramClass>1
</ApplicationControlMacro.ProgramClass>
<ApplicationControlMacro.PSBName>ABCD</ApplicationControlMacro.PSBName>
</ApplicationControlMacro>
</XMI.content>
...
DTD
54
XML to Macro
Parser
52
36
APPLCTN PSB=ABCD,PGMTYPE=(TP,1)

Fig. 10

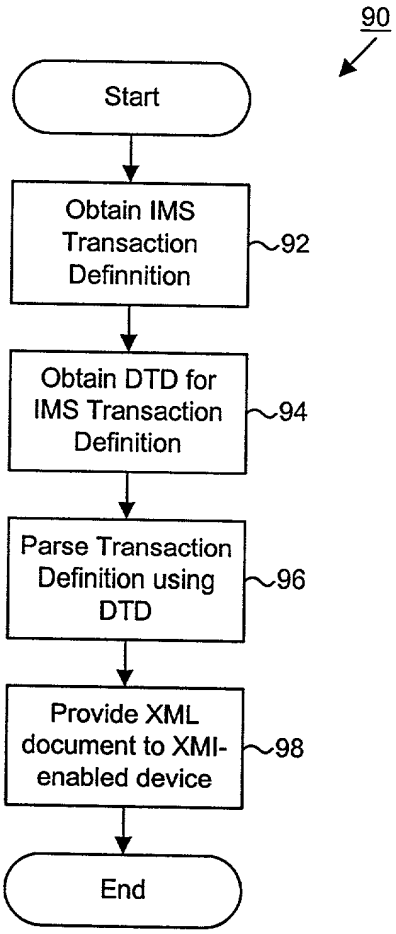
90
Start
Obtain IMS Transaction Definnition
92
Obtain DTD for IMS Transaction Definition
94
Parse Transaction Definition using DTD
96
Provide XML document to XMI-enabled device
98
End

Fig. 11

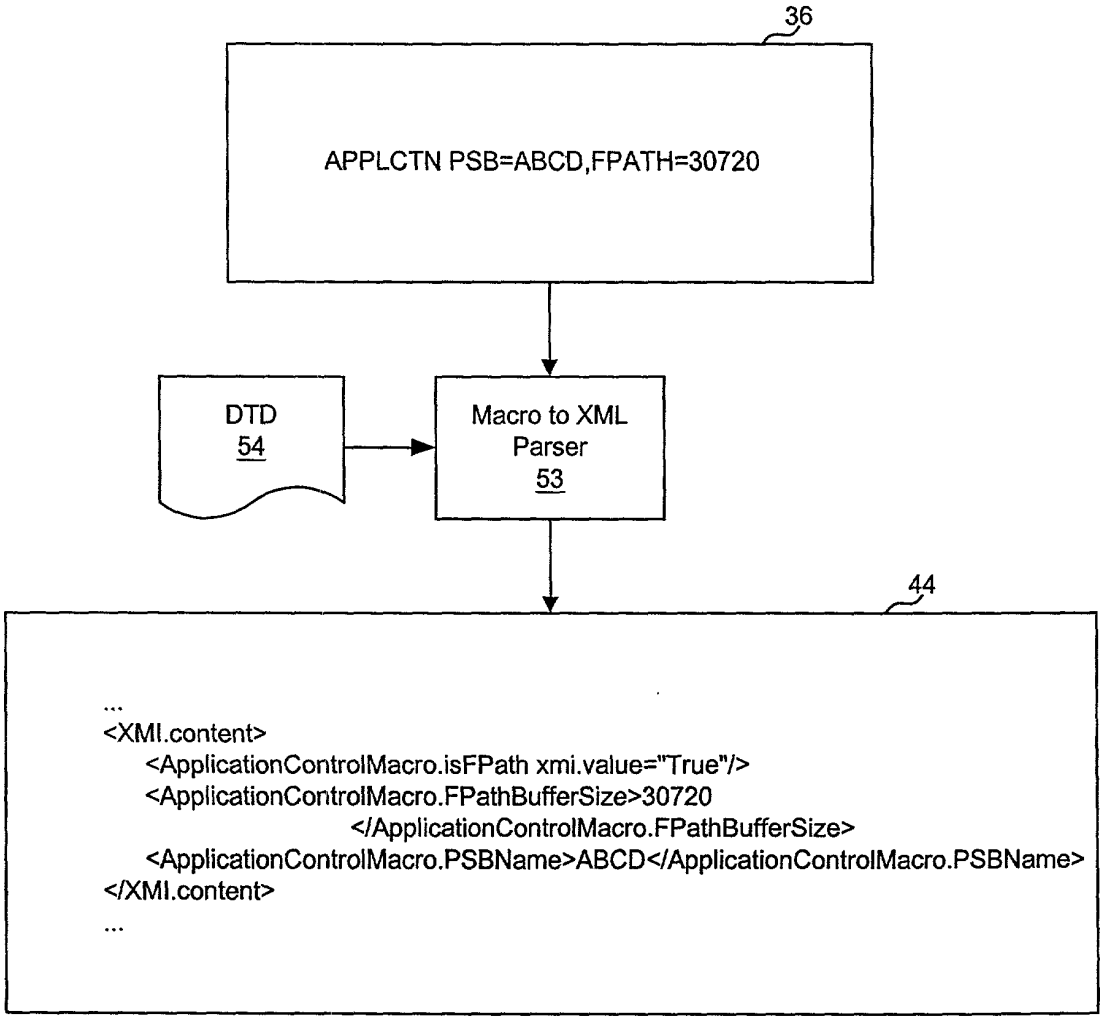
36
APPLCTN PSB=ABCD,FPATH=30720
DTD
54
Macro to XML
Parser
53
44
...
<XMI.content>
<ApplicationControlMacro.isFPath xmi.value="True"/>
<ApplicationControlMacro.FPathBufferSize>30720
</ApplicationControlMacro.FPathBufferSize>
<ApplicationControlMacro.PSBName>ABCD</ApplicationControlMacro.PSBName>
</XMI.content>
...

Fig. 12

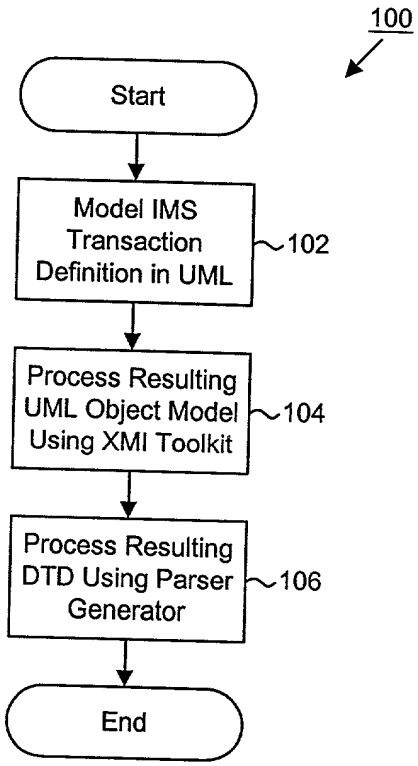
100
Start
Model IMS Transaction Definition in UML
102
Process Resulting UML Object Model Using XMI Toolkit
104
Process Resulting DTD Using Parser Generator
106
End

Fig. 13

Docket Number: ST9-99-145

DECLARATION AND POWER OF ATTORNEY FOR PATENT APPLICATION

As a below named inventor, I hereby declare that:

My residence and citizenship are as stated below next to my name;

I believe I am the original , first and joint inventor of the subject matter which is claimed and for which a patent is sought on the invention entitled

REPRESENTING IMS TRANSACTION DEFINITIONS AS XML DOCUMENTS

the specification of which (check one)

__X__ is attached hereto.
_____ was filed on _____
as Application Serial No. _____
and was amended on _____ (if applicable).

I hereby stat that I have reviewed and understand the contents of the above identified specification, including the claims, as amended by any amendment referred to above.

I acknowledge the duty to disclose information which is material to patentability as defined in Title 37, code of Federal Regulations, Section 1.56.

I hereby claim foreign priority benefits under Title 35, united States Code, Section 119 of any foreign application(s) for patent or inventor's certificate listed below and have also identified below any foreign application for patent or inventor's certificate having a filing date before that of the application on which priority is claimed:

Prior Foreign Application(s)			Priority Claimed
____none____	_____	_____	_____Yes _____No
(Number)	(Country)	(Day/Month/Year filed)	

I hereby claim the benefit under Title 35, Untied States Code, Section 120 of any United States application(s) listed below and, insofar as the subject matter of each of the claims of this application is not disclosed in the prior United States application in the manner provided by the first paragraph of Title 35, United States Code, Section 112, I acknowledge the duty to disclose information which is material to patentability as defined in Title 37, Code of Federal Regulations, Section 1.56, which occurred between the filing date of the prior application and the national or PCT international filing date of this application:

60/151,482	August 30, 1999	Pending
(Application Serial No.)	(Filing Date)	(Status) (patented, pending, abandoned)

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application or any patent issued thereon.

POWER OF ATTORNEY: As named inventor, I hereby appoint the following attorney(s) and/or agent(s) to prosecute this application and transact all business in the Patent and Trademark Office connected therewith. (list name and registration number)

Romualdas Strimaitis:	35,697	Evan R. Witt:	32,512
Timothy M. Farrell:	37,321	A. John Pate:	36,234
Ingrid M. Foerster:	36,511	Gary D.E. Pierce:	38,019
Prentiss W. Johnson:	33,123	David B. Fonda:	39,672
Christopher A. Hughes:	26,914	John R. Thompson:	40,842
John E. Hoel:	26,279	Barton W. Giddings:	41,036
Edward A. Pennington:	32,588	Hal D. Baird:	42,284
Joseph C. Redmond, Jr.:	18,753	Kory D. Christensen:	43,548
Craig J. Madson:	29,407		
L. Craig Metcalf:	31,398		

Send correspondence to:

Kory D. Christensen
MADSON & METCALF
15 West South Temple, Suite 900
Salt Lake City, Utah 84101
Telephone: (801) 537-1700

Full name of sole or first joint-inventor: Shyh-Mei Ho

Inventor's signature: Shyh-Mei F. Ho Date: 4/11/2000

Residence: 10375 Moretti Drive, Cupertino, California 95014

Citizenship: United States of America

Post Office Address: Same

Full name of second joint-inventor: Holger Juschkewitz

Inventor's signature: Date:

Residence: Tuttlinger Str. 110 Stuttgart, Federal Republic of Germany D-70619

Citizenship: German

Post Office Address: Same

Docket Number: ST9-99-145

DECLARATION AND POWER OF ATTORNEY FOR PATENT APPLICATION

As a below named inventor, I hereby declare that:

My residence and citizenship are as stated below next to my name;

I believe I am the original , first and joint inventor of the subject matter which is claimed and for which a patent is sought on the invention entitled

REPRESENTING IMS TRANSACTION DEFINITIONS AS XML DOCUMENTS

the specification of which (check one)

__X__ is attached hereto.
_____ was filed on _____
as Application Serial No. _____
and was amended on _____ (if applicable).

I hereby stat that I have reviewed and understand the contents of the above identified specification, including the claims, as amended by any amendment referred to above.

I acknowledge the duty to disclose information which is material to patentability as defined in Title 37, code of Federal Regulations, Section 1.56.

I hereby claim foreign priority benefits under Title 35, united States Code, Section 119 of any foreign application(s) for patent or inventor's certificate listed below and have also identified below any foreign application for patent or inventor's certificate having a filing date before that of the application on which priority is claimed:

Prior Foreign Application(s)			Priority Claimed
none			_____Yes _____No
(Number)	(Country)	(Day/Month/Year filed)	

I hereby claim the benefit under Title 35, Untied States Code, Section 120 of any United States application(s) listed below and, insofar as the subject matter of each of the claims of this application is not disclosed in the prior United States application in the manner provided by the first paragraph of Title 35, United States Code, Section 112, I acknowledge the duty to disclose information which is material to patentability as defined in Title 37, Code of Federal Regulations, Section 1.56, which occurred between the filing date of the prior application and the national or PCT international filing date of this application:

60/151,482	August 30, 1999	Pending
(Application Serial No.)	(Filing Date)	(Status) (patented, pending, abandoned)

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application or any patent issued thereon.

POWER OF ATTORNEY: As named inventor, I hereby appoint the following attorney(s) and/or agent(s) to prosecute this application and transact all business in the Patent and Trademark Office connected therewith. (list name and registration number)

Romualdas Strimaitis:	35,697	Evan R. Witt:	32,512
Timothy M. Farrell:	37,321	A. John Pate:	36,234
Ingrid M. Foerster:	36,511	Gary D.E. Pierce:	38,019
Prentiss W. Johnson:	33,123	David B. Fonda:	39,672
Christopher A. Hughes:	26,914	John R. Thompson:	40,842
John E. Hoel:	26,279	Barton W. Giddings:	41,036
Edward A. Pennington:	32,588	Hal D. Baird:	42,284
Joseph C. Redmond, Jr.:	18,753	Kory D. Christensen:	43,548
Craig J. Madson:	29,407		
L. Craig Metcalf:	31,398		

Send correspondence to:

Kory D. Christensen
MADSON & METCALF
15 West South Temple, Suite 900
Salt Lake City, Utah 84101
Telephone: (801) 537-1700

Full name of sole or first joint-inventor: Shyh-Mei Ho

Inventor's signature: Date:

Residence: 10375 Moretti Drive, Cupertino, California 95014

Citizenship: United States of America

Post Office Address: Same

Full name of second joint-inventor: Holger Juschkewitz

Inventor's signature: Date: 4/11/2000

Residence: Tuttlinger Str. 110 Stuttgart, Federal Republic of Germany D-70619

Citizenship: German

Post Office Address: Same